

Solaris Volume Manager 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際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェア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	19
1 Solaris Volume Manager の使用開始	25
Solaris Volume Manager のロードマップ – 新機能	26
Solaris Volume Manager のロードマップ – ストレージ容量	26
Solaris Volume Manager のロードマップ – 可用性	28
Solaris Volume Manager のロードマップ – 入出力パフォーマンス	28
Solaris Volume Manager のロードマップ – 管理	29
Solaris Volume Manager のロードマップ – トラブルシューティング	30
2 ストレージ管理の概念	31
ストレージ管理の概要	31
ストレージハードウェア	31
RAID レベル	32
構成の計画に関するガイドライン	33
ストレージの選択	33
一般的なパフォーマンスに関するガイドライン	35
ランダム入出力と順次入出力の最適化	36
ランダム入出力	36
順次アクセス入出力	37
3 Solaris Volume Manager の概要	39
Solaris Volume Manager の新機能	39
Solaris Volume Manager の概要	39
Solaris Volume Manager でのストレージの管理方法	40
Solaris Volume Manager を管理する方法	41
▼ Solaris Volume Manager グラフィカルユーザーインターフェース (GUI) にアクセスす	

る方法	42
Solaris Volume Manager の要件	43
Solaris Volume Manager コンポーネントの概要	43
ボリュームの概要	44
状態データベースと状態データベースの複製	48
ホットスペアプール	49
ディスクセット	49
Solaris Volume Manager 構成に関するガイドライン	50
一般的なガイドライン	50
ファイルシステムに関するガイドライン	51
Solaris Volume Manager コンポーネントの作成の概要	51
Solaris Volume Manager コンポーネントを作成するための前提条件	51
Solaris Volume Manager でのマルチテラバイトのサポートの概要	52
大容量ボリュームのサポートの制限	52
大容量ボリュームの使用	53
Solaris Volume Manager へのアップグレード	53
4 Solaris Volume Manager for Sun Cluster (概要)	55
Solaris Volume Manager for Sun Cluster の概要	55
前提条件: 複数所有者ディスクセット機能に必要なソフトウェアコンポーネント	56
複数所有者ディスクセットの概念	57
複数所有者ディスクセットに関連したタスク	58
Solaris Volume Manager for Sun Cluster 構成	60
複数所有者ディスクセット内の RAID-1 (ミラー) ボリューム	60
複数所有者ディスクセットのミラー所有権	60
データの管理と回復のプロセス	61
5 Solaris Volume Manager の構成と使用 (シナリオ)	63
シナリオの背景情報	63
ハードウェア構成	63
初期の物理ストレージ構成	64
最終的な Solaris Volume Manager 構成	65

6	状態データベース (概要)	67
	Solaris Volume Manager 状態データベースと複製について	67
	多数決アルゴリズムについて	69
	状態データベースの複製の管理	69
	状態データベースの複製のエラー処理	71
	シナリオ - 状態データベースの複製	72
7	状態データベース (タスク)	73
	状態データベースの複製 (タスクマップ)	73
	状態データベースの複製の作成	74
	▼ 状態データベースの複製を作成する方法	74
	状態データベースの複製の保守	76
	▼ 状態データベースの複製のステータスをチェックする方法	76
	▼ 状態データベースの複製を削除する方法	77
8	RAID-0 (ストライプと連結) ボリューム (概要)	79
	RAID-0 ボリュームの概要	79
	RAID-0 (ストライプ) ボリューム	80
	RAID-0 (連結) ボリューム	82
	RAID-0 (連結ストライプ) ボリューム	84
	RAID-0 ボリュームを作成するための背景情報	87
	RAID-0 ボリュームの要件	87
	RAID-0 ボリュームのガイドライン	87
	シナリオ - RAID-0 ボリューム	88
9	RAID-0 (ストライプおよび連結) ボリューム (タスク)	89
	RAID-0 ボリューム (タスクマップ)	89
	RAID-0 (ストライプ) ボリュームの作成	90
	▼ RAID-0 (ストライプ) ボリュームを作成する方法	90
	RAID-0 (連結) ボリュームの作成	92
	▼ RAID-0 (連結) ボリュームを作成する方法	92
	ストレージ容量の拡張	93
	▼ 既存のデータのストレージ容量を拡張する方法	93
	▼ 既存の RAID-0 ボリュームを拡張する方法	95

RAID-0 ボリュームの削除	97
▼ RAID-0 ボリュームを削除する方法	97
10 RAID-1(ミラー)ボリューム(概要)	99
RAID-1(ミラー)ボリュームの概要	99
サブミラーの概要	100
シナリオ - RAID-1(ミラー)ボリューム	100
RAID-1+0 と RAID-0+1 の提供	101
RAID-1 ボリューム(ミラー)の再同期	102
完全な再同期	103
最適化された再同期	103
部分的な再同期	103
RAID-1 ボリュームの作成と保守	104
RAID-1 ボリュームの構成のガイドライン	104
RAID-1 ボリュームのパフォーマンスのガイドライン	105
RAID-1 ボリュームオプションについて	106
保守作業を判断するためのサブミラーのステータスについて	107
RAID-1 ボリュームでシングルユーザーモードでブートする影響	109
シナリオ - RAID-1 ボリューム(ミラー)	110
11 RAID-1(ミラー)ボリューム(タスク)	111
RAID-1 ボリューム(タスクマップ)	111
RAID-1 ボリュームの作成	113
▼ 未使用のスライスから RAID-1 ボリュームを作成する方法	113
▼ ファイルシステムから RAID-1 ボリュームを作成する方法	115
▼ SPARC: ルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法 ...	120
x86: ルート (/) ファイルシステムから RAID-1 ボリュームを作成する	124
ルート (/) ファイルシステムをミラー化した場合のブート時の警告について ...	135
サブミラーに関する作業	136
▼ サブミラーを接続する方法	136
▼ サブミラーを切り離す方法	137
▼ サブミラーをオフラインおよびオンラインにする方法	138
▼ サブミラーの内のスライスを有効にする方法	139
RAID-1 ボリュームの保守	140
▼ ミラーとサブミラーのステータスを表示する方法	140

▼ RAID-1 ボリュームオプションを変更する方法	142
▼ RAID-1 ボリュームを拡張する方法	143
RAID-1 ボリュームのコンポーネント障害に対する処置	144
▼ サブミラー内のスライスを置換する方法	144
▼ サブミラーを置換する方法	146
RAID-1 ボリュームの削除 (ミラー化の解除)	148
▼ ファイルシステムのミラー化を解除する方法	148
▼ アンマウントできないファイルシステムのミラー化を解除する方法	150
RAID-1 ボリューム上でのデータのバックアップ	153
▼ RAID-1 ボリュームのオンラインバックアップを実行する方法	153
12 ソフトパーティション (概要)	157
ソフトパーティションの概要	157
ソフトパーティションの構成のガイドライン	158
シナリオ-ソフトパーティション	159
13 ソフトパーティション (タスク)	161
ソフトパーティション (タスクマップ)	161
ソフトパーティションの作成	162
▼ ソフトパーティションを作成する方法	162
ソフトパーティションの保守	163
▼ ソフトパーティションのステータスをチェックする方法	163
▼ ソフトパーティションを拡張する方法	164
▼ ソフトパーティションを削除する方法	165
14 RAID-5 ボリューム (概要)	167
RAID-5 ボリュームの概要	167
例 - RAID-5 ボリューム	168
例 - 連結 (拡張) RAID-5 ボリューム	169
RAID-5 ボリュームを作成するための背景情報	171
RAID-5 ボリュームの要件	171
RAID-5 ボリュームのガイドライン	172
RAID-5 ボリュームのステータスのチェックの概要	172
RAID-5 ボリューム内のスライスの交換と有効化の概要	174

シナリオ - RAID-5 ボリューム	175
15 RAID-5 ボリューム (タスク)	177
RAID-5 ボリューム (タスクマップ)	177
RAID-5 ボリュームの作成	178
▼ RAID-5 ボリュームを作成する方法	178
RAID-5 ボリュームの保守	179
▼ RAID-5 ボリュームのステータスをチェックする方法	179
▼ RAID-5 ボリュームを拡張する方法	180
▼ RAID-5 ボリューム内のコンポーネントを有効にする方法	182
▼ RAID-5 ボリューム内のコンポーネントを置き換える方法	183
16 ホットスペアプール (概要)	187
ホットスペアとホットスペアプールの概要	187
ホットスペア	188
ホットスペアプール	188
ホットスペアのしくみ	189
ホットスペアプールの状態	189
例 - ホットスペアプール	190
シナリオ - ホットスペア	191
17 ホットスペアプール (タスク)	193
ホットスペアプール (タスクマップ)	193
ホットスペアプールの作成	194
▼ ホットスペアプールを作成する方法	194
▼ ホットスペアプールにスライスを追加する方法	195
ホットスペアプールのボリュームへの関連付け	197
▼ ホットスペアプールをボリュームに関連付ける方法	197
▼ 関連付けられているホットスペアプールを変更する方法	198
ホットスペアプールの保守	200
▼ ホットスペアとホットスペアプールのステータスを確認する方法	200
▼ ホットスペアプール内のホットスペアを置き換える方法	200
▼ ホットスペアプールからホットスペアを削除する方法	202
▼ ホットスペアを有効にする方法	203

18	ディスクセット(概要)	205
	ディスクセットの新機能	205
	ディスクセットの概要	205
	ディスクセットのタイプ	206
	ローカルディスクセット	206
	名前付きディスクセット	206
	Solaris Volume Manager ディスクセット管理	208
	ディスクセットの予約	209
	ディスクセットの解放	210
	ディスクセットのインポート	210
	自動ディスクパーティション分割	211
	ディスクセット名の要件	213
	例—2つの共有ディスクセット	214
	ディスクセットの操作のガイドライン	215
	ディスクセット内の非同期共有ストレージ	216
	シナリオ-ディスクセット	216
19	ディスクセット(タスク)	217
	ディスクセット(タスクマップ)	217
	ディスクセットの作成	219
	▼ディスクセットを作成する方法	219
	ディスクセットの拡張	220
	▼ディスクセットにディスクを追加する方法	220
	▼ディスクセットに別のホストを追加する方法	222
	▼ディスクセット内に Solaris Volume Manager コンポーネントを作成する方法	223
	ディスクセットの保守	224
	▼ディスクセットのステータスを確認する方法	224
	▼ディスクセットからディスクを削除する方法	225
	▼ディスクセットを取得する方法	226
	▼ディスクセットを解放する方法	228
	▼ホストまたはディスクセットを削除する方法	229
	ディスクセットのインポート	231
	▼インポートに使用可能なディスクセットに関するレポートを出力する方法	231
	▼ディスクセットをあるシステムから別のシステムにインポートする方法	232

20 Solaris Volume Manager の保守 (タスク)	235
Solaris Volume Manager の保守 (タスクマップ)	235
Solaris Volume Manager 構成の表示	236
▼ Solaris Volume Manager ボリューム構成を表示する方法	236
次に進む手順	239
ボリュームの名前の変更	240
ボリュームの名前の変更の背景情報	240
ボリューム名の交換	241
▼ ボリュームの名前を変更する方法	241
構成ファイルの操作	243
▼ 構成ファイルを作成する方法	243
▼ 構成ファイルから Solaris Volume Manager を初期化する方法	243
Solaris Volume Manager のデフォルト値の変更	245
growfs コマンドを使用したファイルシステムの拡張	246
スライスとボリュームの拡張の背景情報	246
▼ ファイルシステムを拡張する方法	247
RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要	248
コンポーネントの有効化	249
コンポーネントを別の使用可能なコンポーネントに置き換える	249
「保守」状態と「最後にエラー」状態	250
RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の背景 情報	251
21 Solaris Volume Manager のベストプラクティス	253
小規模なサーバーの配備	253
ネットワーク接続されたストレージデバイスによる Solaris Volume Manager の使 用	255
22 トップダウンボリューム作成 (概要)	257
トップダウンボリューム作成の概要	257
ディスクセットによるトップダウンボリューム作成の実装	258
トップダウンボリューム作成のプロセス	259
トップダウンボリューム作成に使用できるディスクの判別	260

23	ボリュームのトップダウン作成(タスク)	261
	ボリュームのトップダウン作成(タスクマップ)	261
	ボリュームをトップダウン作成するための前提条件	262
	ボリュームの自動作成	263
	出力の詳細度指定によるボリューム作成分析	263
	▼ metassist コマンドを使用して RAID-1(ミラー) ボリュームを作成する方法	264
	metassist コマンドによるファイルベースのデータ処理	267
	metassist コマンドによるコマンドファイル(シェルスクリプト)の作成	267
	▼ metassist コマンドを使用してコマンドファイル(シェルスクリプト)を作成する 方法	267
	metassist コマンドで作成されたシェルスクリプトによるボリュームの作成 ...	271
	metassist コマンドによるボリューム構成ファイルの作成	272
	▼ metassist コマンドを使用してボリューム構成ファイルを作成する方法	272
	metassist コマンドのデフォルト動作の変更	274
	ボリュームデフォルトファイルの変更	275
24	モニタリングとエラー報告(タスク)	277
	Solaris Volume Manager のモニタリングと報告(タスクマップ)	278
	エラーを周期的にチェックするための mdmonitord コマンドの構成	278
	▼ mdmonitord コマンドを構成してエラーを周期的にチェックする方法	278
	Solaris Volume Manager SNMP エージェントの概要	279
	Solaris Volume Manager SNMP エージェントの構成	280
	▼ Solaris Volume Manager SNMP エージェントを構成する方法	280
	Solaris Volume Manager SNMP エージェントの制限	282
	cron ジョブによる Solaris Volume Manager のモニタリング	283
	▼ ボリュームのエラーを自動的にチェックする方法	283
25	Solaris Volume Manager のトラブルシューティング(タスク)	291
	Solaris Volume Manager のトラブルシューティング(タスクマップ)	292
	システムのトラブルシューティングの概要	293
	システムのトラブルシューティングの前提条件	293
	Solaris Volume Manager のトラブルシューティングの一般的なガイドライン	293
	一般的なトラブルシューティング方法	293
	ディスクの交換	294
	▼ 不良ディスクを交換する方法	294

ディスク移動の問題からの回復	296
ディスク移動とデバイス ID の概要	297
名前のないデバイスに関するエラーメッセージの解決	297
Solaris 10 リリースにアップグレードしたあとのデバイス ID の不一致	298
ブートの問題からの回復	300
ブートの問題の背景情報	300
/etc/vfstab 内の不適切なエントリを修正する方法	301
▼ ルート (/) RAID-1 (ミラー) ボリュームを回復する	301
▼ ブートデバイスの障害から回復する方法	303
状態データベース複製の障害からの回復	307
▼ 状態データベースの複製数の不足から回復する方法	307
ソフトパーティションの問題からの回復	310
▼ ソフトパーティションの構成データを回復する方法	310
別のシステムからのストレージの回復	312
▼ ローカルディスクセットからストレージを回復する方法	312
既知のディスクセットからのストレージの回復	316
ディスクセットの問題からの回復	319
ディスクセットの所有権を取得できないときには	319
ufsdump コマンドによるマウント済みファイルシステムのバックアップの実行	320
▼ RAID-1 ボリューム上のマウント済みファイルシステムのバックアップを実行する 方法	320
システム回復の実行	322
▼ Solaris Volume Manager 構成を使用してシステムを回復する方法	322
A 重要な Solaris Volume Manager ファイル	325
システムファイルとスタートアップファイル	325
手動で構成するファイル	326
md.tab ファイルの概要	326
B Solaris Volume Manager のクイックリファレンス	329
コマンド行のリファレンス	329
C Solaris Volume Manager CIM/WBEM API	331
Solaris Volume Manager の管理	331

索引 333

目次

図 3-1	Solaris 管理コンソールでの拡張ストレージツール (Solaris Volume Manager) のビュー	42
図 3-2	ボリューム、物理ディスク、およびスライスの関係	46
図 4-1	サンプルのクラスタ構成	56
図 5-1	ストレージのシナリオの基本的なハードウェア図	64
図 8-1	RAID-0 (ストライプ) ボリュームの例	82
図 8-2	RAID-0 (連結) ボリュームの例	84
図 8-3	RAID-0 (連結ストライプ) ボリュームの例	86
図 10-1	RAID-1 (ミラー) の例	101
図 10-2	RAID-1+0 の例	102
図 14-1	RAID-5 ボリュームの例	169
図 14-2	拡張された RAID-5 ボリュームの例	170
図 16-1	ホットスペアプールの例	191
図 18-1	ディスクセットの例	214
図 21-1	小規模なシステム構成	254
図 22-1	トップダウンボリューム作成の処理のオプション	259

表目次

表 2-1	ストレージのタイプの比較	34
表 2-2	冗長ストレージの最適化	34
表 3-1	Solaris Volume Manager 機能のサマリー	44
表 3-2	ボリュームのクラス	45
表 10-1	RAID-1 ボリュームの読み取りポリシー	107
表 10-2	RAID-1 ボリュームの書き込みポリシー	107
表 10-3	サブミラーの状態	108
表 10-4	サブミラーのスライスの状態	109
表 14-1	RAID-5 ボリュームの状態	173
表 14-2	RAID-5 スライスの状態	173
表 16-1	ホットスペアプールの状態(コマンド行)	190
表 18-1	ディスクセットのボリューム名の例	214
表 25-1	Solaris Volume Manager での一般的なブートの問題	300
表 B-1	Solaris Volume Manager コマンド	329

はじめに

『Solaris Volume Manager 管理ガイド』では、Solaris Volume Manager を使用してシステムのストレージニーズを管理する方法について説明します。Solaris Volume Manager を使用すると、RAID-0(連結およびストライプ)ボリューム、RAID-1(ミラー)ボリューム、RAID-5 ボリューム、およびソフトパーティションを作成、変更、および使用することができます。

注 - この Oracle Solaris のリリースでは、SPARC および x86 系列のプロセッサアーキテクチャーを使用するシステムをサポートしています。サポートされるシステムは、Oracle Solaris OS: Hardware Compatibility Lists に記載されています。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書の x86 に関連する用語については、次を参照してください。

- x86 は、64 ビットおよび 32 ビットの x86 互換製品系列を指します。
- x64 は特に 64 ビット x86 互換 CPU を指します。
- 「32 ビット x86」は、x86 をベースとするシステムに関する 32 ビット特有の情報を指します。

サポートされるシステムについては、[Oracle Solaris OS: Hardware Compatibility Lists](#) を参照してください。

対象読者

システムおよびストレージ管理者は、このドキュメントを使用して次の内容を識別できます。

- Solaris Volume Manager によってサポートされるタスク
- Solaris Volume Manager を使用して、より信頼性の高いアクセス可能なデータを提供する方法

内容の紹介

『Solaris Volume Manager 管理ガイド』の内容は次のとおりです。

第1章「Solaris Volume Manager の使用開始」では、このドキュメントで説明されている概念とタスクへの詳細な「ロードマップ」について説明します。この章は、このドキュメント内を移動する場合の手助けとして使用してください。

第2章「ストレージ管理の概念」では、このテクノロジーにまだ精通していない読者のために、ストレージ管理の一般的な概念を紹介します。

第3章「Solaris Volume Manager の概要」では、Solaris Volume Manager について説明します。この章では、製品に関連する重要な概念を紹介するとともに、Solaris Volume Manager ツールにアクセスする方法について説明します。

第4章「Solaris Volume Manager for Sun Cluster (概要)」では、複数所有者ディスクセットを紹介します。複数所有者ディスクセットによって、Sun クラスタ環境での Solaris Volume Manager の使用が拡張されます。

第5章「Solaris Volume Manager の構成と使用(シナリオ)」では、このドキュメント全体にわたって使用されるストレージ構成シナリオについて説明します。このシナリオは、ユーザーが Solaris Volume Manager 製品を理解するのに役立つことを目的にしています。

第6章「状態データベース(概要)」では、状態データベースと状態データベースの複製に関連した概念について説明します。

第7章「状態データベース(タスク)」では、状態データベースと状態データベースの複製に関連したタスクを実行する方法について説明します。

第8章「RAID-0(ストライプと連結)ボリューム(概要)」では、RAID-0(ストライプおよび連結)ボリュームに関連した概念について説明します。

第9章「RAID-0(ストライプおよび連結)ボリューム(タスク)」では、RAID-0(ストライプおよび連結)ボリュームに関連したタスクを実行する方法について説明します。

第10章「RAID-1(ミラー)ボリューム(概要)」では、RAID-1(ミラー)ボリュームに関連した概念について説明します。

第11章「RAID-1(ミラー)ボリューム(タスク)」では、RAID-1(ミラー)ボリュームに関連したタスクを実行する方法について説明します。

第12章「ソフトパーティション(概要)」では、Solaris Volume Manager のソフトパーティション機能に関連した概念について説明します。

第13章「ソフトパーティション(タスク)」では、ソフトパーティションに関連したタスクを実行する方法について説明します。

第14章「RAID-5 ボリューム(概要)」では、RAID-5 ボリュームに関連した概念について説明します。

第15章「RAID-5ボリューム(タスク)」では、RAID-5ボリュームに関連したタスクを実行する方法について説明します。

第16章「ホットスペアプール(概要)」では、ホットスペアとホットスペアプールに関連した概念について説明します。

第17章「ホットスペアプール(タスク)」では、ホットスペアとホットスペアプールに関連したタスクを実行する方法について説明します。

第18章「ディスクセット(概要)」では、ディスクセットに関連した概念について説明します。

第19章「ディスクセット(タスク)」では、ディスクセットに関連したタスクを実行する方法について説明します。

第20章「Solaris Volume Managerの保守(タスク)」では、特定のSolaris Volume Managerコンポーネントには関連していない、いくつかの一般的な保守タスクについて説明します。

第21章「Solaris Volume Managerのベストプラクティス」では、Solaris Volume Managerの構成と使用に関するいくつかの「ベストプラクティス」情報について説明します。

第23章「ボリュームのトップダウン作成(タスク)」では、Solaris Volume Managerのトップダウンボリューム作成機能の概念と、それに関連したタスクについて説明します。

第24章「モニタリングとエラー報告(タスク)」では、Solaris Volume Manager SNMPエージェントの使用や、その他のエラーチェックアプローチのための概念と手順について説明します。

第25章「Solaris Volume Managerのトラブルシューティング(タスク)」では、Solaris Volume Manager環境での一般的な問題のトラブルシューティングと解決に関する情報について説明します。

付録A「重要なSolaris Volume Managerファイル」では、重要なSolaris Volume Managerファイルの一覧を示します。

付録B「Solaris Volume Managerのクイックリファレンス」では、コマンドやその他の役立つ情報を要約した表を示します。

付録C「Solaris Volume Manager CIM/WBEM API」では、WBEMに準拠した管理ツールからのオープンなSolaris Volume Manager管理を可能にするCIM/WBEM APIを簡単に紹介します。

関連ドキュメント

Solaris Volume Manager は、Solaris オペレーティングシステムで使用可能ないくつかのシステム管理ツールの 1 つです。システム管理の全体的な特徴や機能および関連するツールについては、次のドキュメントを参照してください。

- 『Oracle Solaris の管理: 基本管理』
- 『Solaris のシステム管理 (上級編)』
- 『Oracle Solaris の管理: デバイスとファイルシステム』

Oracle サポートへのアクセス

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

表記上の規則

次の表では、このドキュメントで使用される表記上の規則について説明します。

表 P-1 表記上の規則

字体	説明	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、コード例を示します。	<code>.login</code> ファイルを編集します。 <code>ls -a</code> を使用してすべてのファイルを表示します。 <code>machine_name% you have mail.</code>
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。	<code>machine_name% su</code> <code>Password:</code>
<i>aabbcc123</i>	プレースホルダ: 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には、 <code>rm filename</code> と入力します。

表 P-1 表記上の規則 (続き)

字体	説明	例
<i>AaBbCc123</i>	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るコピーです。 ファイルを保存しないでください。 注: いくつか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

Oracle Solaris OS に含まれるシェルで使用する、UNIX のシステムプロンプトとスーパーユーザープロンプトを次に示します。コマンド例のシェルプロンプトから、通常ユーザーと特権ユーザーのどちらがコマンドを実行すべきかがわかります。

表 P-2 シェルプロンプト

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$
Bash シェル、Korn シェル、および Bourne シェルのスーパーユーザー	#
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

Solaris Volume Manager の使用開始

『Solaris Volume Manager 管理ガイド』では、Solaris Volume Manager を使用してシステムを設定および保守することによりストレージの高可用性、柔軟性、および信頼性を管理する方法について説明します。

この章は、ストレージ容量の設定などの特定の Solaris Volume Manager タスクに関する情報を見つけるための高レベルのガイドとして機能します。この章では、Solaris Volume Manager を使用するために必要になるすべてのタスクを扱っているわけではありません。代わりに、この章では新機能の概要や、Solaris Volume Manager の概念に関連した一般的なタスクを説明している手順を簡単に見つける方法を示します。

この章には、次のロードマップが含まれています。

- 26 ページの「Solaris Volume Manager のロードマップ – 新機能」
- 26 ページの「Solaris Volume Manager のロードマップ – ストレージ容量」
- 28 ページの「Solaris Volume Manager のロードマップ – 可用性」
- 28 ページの「Solaris Volume Manager のロードマップ – 入出力パフォーマンス」
- 29 ページの「Solaris Volume Manager のロードマップ – 管理」
- 30 ページの「Solaris Volume Manager のロードマップ – トラブルシューティング」



注意 - Solaris Volume Manager を正しく使用しないと、データが破棄される場合があります。Solaris Volume Manager では、ディスクやディスク上のデータを確実に管理するための強力な方法が提供されます。ただし、特にアクティブな Solaris Volume Manager 構成を変更する前には、データのバックアップを常に保持するようにしてください。

Solaris Volume Manager のロードマップ — 新機能

タスク	説明	参照先
1つまたは複数のコンポーネントが 1T バイトを超えるストレージを管理する	1T バイトを超えるサイズの物理的な論理ユニット番号 (LUN) を使用するか、または 1T バイトを超える論理ボリュームを作成します。	52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」
ディスクセットをあるシステムから別のシステムにインポートする	異なるシステム上に作成されているディスクセットであっても、ディスクセットをインポートするには <code>metaimport</code> コマンドを使用します。このコマンドは、拡張されたデバイス ID サポートを使用して、指定されたディスクセット内のディスク移動を自動的に追跡します。	210 ページの「ディスクセットのインポート」 216 ページの「ディスクセット内の非同期共有ストレージ」
複数所有者ディスクセットを作成および管理する	Sun Cluster 環境内の複数所有者ディスクセットを管理するには、 <code>metaset -M</code> を使用します。	58 ページの「複数所有者ディスクセットに関連したタスク」

Solaris Volume Manager のロードマップ — ストレージ容量

タスク	説明	参照先
ストレージを設定する	RAID-0 または RAID-5 ボリュームを作成することによって、スライスにまたがるストレージを作成します。そのあと、その RAID-0 または RAID-5 ボリュームを、 <code>raw</code> デバイスにアクセスするファイルシステムまたは任意のアプリケーション (データベースなど) で使用できます。	90 ページの「RAID-0 (ストライプ) ボリュームを作成する方法」 92 ページの「RAID-0 (連結) ボリュームを作成する方法」 113 ページの「未使用のスライスから RAID-1 ボリュームを作成する方法」 115 ページの「ファイルシステムから RAID-1 ボリュームを作成する方法」 178 ページの「RAID-5 ボリュームを作成する方法」

タスク	説明	参照先
既存のファイルシステムを拡張する	RAID-0(連結) ボリュームを作成したあと、そのボリュームにスライスを追加することによって、既存のファイルシステムの容量を増やします。	93 ページの「既存のデータのストレージ容量を拡張する方法」
既存の RAID-0(連結またはストライプ) ボリュームを拡張する	既存の RAID-0 ボリュームに追加のスライスを連結することによって、そのボリュームを拡張します。	95 ページの「既存の RAID-0 ボリュームを拡張する方法」
RAID-5 ボリュームを拡張する	RAID-5 ボリュームに追加のスライスを連結することによって、そのボリュームの容量を拡張します。	180 ページの「RAID-5 ボリュームを拡張する方法」
拡張されたボリューム上の UFS ファイルシステムのサイズを増やす	growfs コマンドを使用して、マウント中の UFS のサイズをデータアクセスを中断することなく拡張することによって、ファイルシステムを拡張します。	247 ページの「ファイルシステムを拡張する方法」
8 スライスのハードパーティションの制限を超えて、スライスまたは論理ボリュームをより小さなパーティションに再分割する	ソフトパーティションを使用して、論理ボリュームまたはスライスを再分割します。	162 ページの「ソフトパーティションを作成する方法」
ファイルシステムを作成する	RAID-0(ストライプまたは連結)、RAID-1(ミラー)、RAID-5、またはソフトパーティション上にファイルシステムを作成します。	『Oracle Solaris の管理: デバイスとファイルシステム』の第 15 章「ファイルシステムの作成およびマウント(タスク)」

Solaris Volume Manager のロードマップー可用性

タスク	説明	参照先
データ可用性を最大化する	データの複数のコピーを保持するには、Solaris Volume Manager のミラー化機能を使用します。データの準備として未使用のスライスから RAID-1 ボリュームを作成するか、または既存のファイルシステム (ルート (/) や /usr を含む) をミラー化することができます。	113 ページの「未使用のスライスから RAID-1 ボリュームを作成する方法」 115 ページの「ファイルシステムから RAID-1 ボリュームを作成する方法」
最小のハードウェアコストでデータ可用性を向上させる	Solaris Volume Manager の RAID-5 ボリュームを使用して、最小限のハードウェアでデータ可用性を向上させます。	178 ページの「RAID-5 ボリュームを作成する方法」
既存の RAID-1 または RAID-5 ボリュームのデータ可用性を向上させる	ホットスワップを作成し、それを RAID-1 ボリュームまたは RAID-5 ボリュームのサブミラーに関連付けることによって、RAID-1 または RAID-5 ボリュームのデータ可用性を向上させます。	194 ページの「ホットスワップの作成」 197 ページの「ホットスワップのボリュームへの関連付け」

Solaris Volume Manager のロードマップー入出力パフォーマンス

タスク	説明	参照先
RAID-1 ボリュームの読み取りおよび書き込みポリシーを調整する	特定の構成の入出力パフォーマンスを向上させるには、RAID-1 ボリュームの読み取りおよび書き込みポリシーを指定します。	106 ページの「RAID-1 ボリュームの読み取り/書き込みポリシー」 142 ページの「RAID-1 ボリュームオプションを変更する方法」

タスク	説明	参照先
デバイスパフォーマンスを最適化する	ストライプを構成するデバイスの入出力パフォーマンスを最適化するには、RAID-0(ストライプ)ボリュームを作成します。ランダムアクセスまたは順次アクセスのためのインターレース値を最適化できます。	90 ページの「RAID-0(ストライプ)ボリュームの作成」
RAID-0(ストライプ)内のデバイスパフォーマンスを維持する	容量が不足しているストライプまたは連結を、そこに新しいコンポーネントを連結することによって拡張します。入出力パフォーマンスを向上させるには、スライスの連結よりストライプの連結の方が適しています。	93 ページの「ストレージ容量の拡張」

Solaris Volume Manager のロードマップー管理

タスク	説明	参照先
ボリューム管理構成をグラフィカルに管理する	ボリューム管理構成を管理するには、Solaris 管理コンソールのグラフィカルユーザーインターフェース (GUI) を使用します。	Solaris 管理コンソールアプリケーションの Solaris Volume Manager (拡張ストレージ) ノード内からのオンラインヘルプ
スライスとファイルシステムをグラフィカルに管理する	ディスクのパーティション分割や UFS ファイルシステムの構成などのタスクを実行してディスクやファイルシステムを管理するには、Solaris 管理コンソールの GUI を使用します。	Solaris 管理コンソールアプリケーション内からのオンラインヘルプ
Solaris Volume Manager を最適化する	Solaris Volume Manager のパフォーマンスは、適切に設計された構成に依存します。構成は、作成したあとにモニタリングや調整を行う必要があります。	50 ページの「Solaris Volume Manager 構成に関するガイドライン」 243 ページの「構成ファイルの操作」
将来の拡張計画	ファイルシステムは容量が不足する傾向にあるため、ファイルシステムを連結に配置することによって将来の拡張を計画できます。	92 ページの「RAID-0(連結)ボリュームの作成」 93 ページの「ストレージ容量の拡張」

Solaris Volume Manager のロードマップ—トラブルシューティング

タスク	説明	参照先
障害が発生しているスライスを交換する	ディスクに障害が発生した場合は、Solaris Volume Manager 構成で使用されているスライスを交換する必要があります。RAID-0 ボリュームの場合は、新しいスライスを使用し、ボリュームを削除して再作成してから、バックアップからデータを復元する必要があります。RAID-1 および RAID-5 ボリューム内のスライスを、データを失うことなく交換して再同期することができます。	144 ページの「RAID-1 ボリュームのコンポーネント障害に対する処置」 183 ページの「RAID-5 ボリューム内のコンポーネントを置き換える方法」
ブートの問題から回復する	システムをブートするとき、ハードウェアの問題やオペレータのエラーのために特殊な問題が発生する場合があります。	301 ページの「/etc/vfstab 内の不適切なエントリを修正する方法」 307 ページの「状態データベースの複製数の不足から回復する方法」 303 ページの「ブートデバイスの障害から回復する方法」

ストレージ管理の概念

この章では、ストレージ管理のいくつかの一般的な概念を簡単に紹介します。

この章では、次の内容について説明します。

- 31 ページの「ストレージ管理の概要」
- 33 ページの「構成の計画に関するガイドライン」
- 35 ページの「一般的なパフォーマンスに関するガイドライン」
- 36 ページの「ランダム入出力と順次入出力の最適化」

ストレージ管理の概要

ストレージの管理をどのように選択するかによって、システム上のアクティブなデータを格納するデバイスの制御方法が決定されます。アクティブなデータを役立てるには、ハードウェア障害やソフトウェア障害などの予期しないイベントのあとでも使用可能であり、持続的な状態が維持されるようにする必要があります。

ストレージハードウェア

データを格納できるデバイスには、さまざまなものがあります。ストレージニーズを最適に満たすためのデバイスの選択は、主に次の3つの要因によって決まります。

- パフォーマンス
- 可用性
- コスト

Solaris Volume Manager を使用すると、パフォーマンス、可用性、およびコストの間のトレードオフを管理しやすくなります。多くの場合は、Solaris Volume Manager によってトレードオフの多くを軽減できます。

Solaris Volume Manager は、Solaris オペレーティングシステムを実行する任意のシステム上のサポートされる任意のストレージとともに適切に動作します。

RAID レベル

RAID は、Redundant Array of Inexpensive (または Independent) Disks の頭字語です。RAID は、ユーザーには1つの大容量ディスクドライブに見える、アレイまたは「ボリューム」と呼ばれる一連のディスクを指します。構成に応じて、このアレイは信頼性、応答時間、またはストレージ容量を向上させます。

技術的には、0 から 5 までの 6 つの RAID レベルがあります。各レベルは、データの冗長性を確保しながら、データを分散させるための方法を示します。(RAID レベル 0 はデータの冗長性を提供しませんが、通常は RAID の分類として含まれます。RAID レベル 0 は、使用されているほとんどの RAID 構成の基礎を提供します。)RAID レベル 2、3、および 4 をサポートしているストレージ環境は非常に少ないため、ここではこれらの環境について説明していません。

Solaris Volume Manager は、次の RAID レベルをサポートしています。

- **RAID レベル 0** - ストライプと連結は冗長性を提供しませんが、これらのボリュームは通常 RAID-0 と呼ばれます。データは基本的に、複数の物理ディスクにわたって交互にかつ均等に割り当てられた、比較的小さな、均一のサイズのフラグメントにわたって分散されます。どの 1 台のドライブに障害が発生しても、データが失われる場合があります。RAID-0 は、高いデータ転送速度と高い入出力スループットを提供しますが、1 つのディスクに比べて信頼性と可用性は低下します。
- **RAID レベル 1** - ミラー化は、等しい量のディスク容量を使用して、データやデータのコピー (ミラー) を格納します。データは、2 台以上の物理ディスクにわたって複製またはミラー化されます。両方のドライブから同時にデータを読み取ることができます。つまり、どちらのドライブも任意のリクエストを処理できるため、パフォーマンスが向上します。1 つの物理ディスクに障害が発生しても、パフォーマンスの低下やデータの損失なしにミラーを引き続き使用できます。

Solaris Volume Manager は、基となるボリュームに応じて、RAID-0+1 と (透過的に) RAID-1+0 ミラー化の両方をサポートします。詳細は、[101 ページの「RAID-1+0 と RAID-0+1 の提供」](#)を参照してください。

- **RAID レベル 5** - RAID-5 は、ストライプ化を使用して、データをアレイ内のディスクにわたって分散させます。RAID-5 はまた、ある程度のデータの冗長性を提供するためのパリティ情報も記録します。RAID-5 ボリュームは、障害を発生させることなく、基となるデバイスのエラーに耐えることができます。RAID-5 ボリュームがホットスペアと組み合わせて使用された場合、そのボリュームは障害を発生させることなく、複数のエラーに耐えることができます。RAID-5 ボリュームは、障害が発生したデバイスで動作している場合、パフォーマンスが大幅に低下します。

RAID-5モデルでは、すべてのデバイスに、パリティーストライプを含む1つの領域と、データを含むその他の領域があります。パリティはアレイ内のすべてのディスクにわたって分散されるため、書き込み時間が短縮されます。書き込み時間が短縮されるのは、専用のパリティディスクがデータを受け入れ可能になるまで書き込みを待たせる必要がないためです。

構成の計画に関するガイドライン

ストレージ管理の構成を計画している場合は、特定のどの構成でも、パフォーマンス、可用性、およびハードウェアコストの間のトレードオフがあることに注意してください。実際の構成にとって何が最適に機能するかを判断するために、各種の変数で実験することが必要になる場合があります。

このセクションでは、次のタイプのボリュームを使用する場合のガイドラインについて説明します。

- RAID-0(連結およびストライプ)ボリューム
- RAID-1(ミラー)ボリューム
- RAID-5ボリューム
- ソフトパーティション
- Solaris Volume Manager ボリューム上に構築されたファイルシステム

ストレージの選択

ストレージ管理アプローチを実装する前に、どのような種類のストレージデバイスを使用するかを決定する必要があります。この一連のガイドラインでは、その選択に役立つように、ストレージのさまざまなタイプを比較します。追加の一連のガイドラインは、Solaris Volume Manager で実装される特定のタイプのストレージに適用されます。詳細は、各ボリュームタイプに関する特定の章を参照してください。

注-ここに一覧表示されているストレージのタイプは、相互に排他的ではありません。複数の目標を満たすために、これらのボリュームを組み合わせる使用できます。たとえば、最初に、冗長性のためにRAID-1ボリュームを作成することができます。次に、追加できる個別のファイルシステムの数を増やすために、そのRAID-1ボリューム上にソフトパーティションを作成することができます。

次の表は、各タイプのストレージで使用可能な機能の比較を示しています。

表2-1 ストレージのタイプの比較

要件	RAID-0(連結)	RAID-0(ストライプ)	RAID-1(ミラー)	RAID-5	ソフトパーティション
データの冗長性	いいえ	いいえ	はい	はい	いいえ
読み取りパフォーマンスの向上	いいえ	はい	基となるデバイスによって異なる	はい	いいえ
書き込みパフォーマンスの向上	いいえ	はい	いいえ	いいえ	いいえ
デバイスあたり8つを超えるスライス	いいえ	いいえ	いいえ	いいえ	はい
使用可能なストレージスペースの大容量化	はい	はい	いいえ	はい	いいえ

次の表は、RAID-1 ボリュームと RAID-5 ボリュームでの書き込み操作、ランダム読み取り、およびハードウェアコストの間のトレードオフの概要を示しています。

表2-2 冗長ストレージの最適化

	RAID-1(ミラー)	RAID-5
書き込み操作	高速	低速
ランダム読み取り	高速	低速
ハードウェアコスト	高い	低い

次の一覧は、これらの表に概要が示されている情報を要約したものです。

- RAID-0 ボリューム (ストライプおよび連結) とソフトパーティションでは、データの冗長性は提供されません。
- 連結は、小さなランダム入出力操作に対して適切に機能します。
- ストライプ化は、大きな順次入出力操作やランダム入出力操作に対して適切に機能します。
- ミラー化によって読み取りパフォーマンスは向上する可能性があります、ミラーでは書き込みパフォーマンスが常に低下します。

- RAID-5 ボリュームには読み取り/変更/書き込みの性質があるため、書き込みが20%を超えるボリュームをRAID-5にすべきではありません。冗長性が必要な場合は、ミラー化を検討してください。
- RAID-5 の書き込みはミラー化された書き込みほど高速ではなく、そのミラー化された書き込みも、保護されていない書き込みほど高速ではありません。
- ソフトパーティションは、非常に大容量のストレージデバイスを管理するために役立ちます。

注- これらの汎用のストレージオプションに加えて、Solaris Volume Manager を使用して冗長なデバイスをサポートする方法の詳細は、[49 ページの「ホットスベアプール」](#)を参照してください。

一般的なパフォーマンスに関するガイドライン

ストレージ構成を設計する場合は、次のパフォーマンスに関するガイドラインを考慮してください。

- ストライプ化は一般に、最高のパフォーマンスを発揮しますが、ストライプ化によってデータの冗長性は提供されません。大量の書き込みを行うアプリケーションでは、一般に、RAID-5 ボリュームより RAID-1 ボリュームの方がパフォーマンスは高くなります。
- RAID-1 ボリュームと RAID-5 ボリュームはどちらもデータ可用性を向上させますが、一般に、どちらのタイプのボリュームも書き込み操作のパフォーマンスは低くなります。ミラー化によって、ランダム読み取りパフォーマンスが向上します。
- RAID-5 ボリュームは RAID-1 ボリュームに比べてハードウェアコストが低い一方、RAID-0 ボリュームには追加のハードウェアコストがありません。
- ストライプと RAID-5 ボリュームはどちらも、データを複数のディスクドライブにわたって分散させるため、入出力負荷の分散に役立ちます。
- もっとも頻繁にアクセスされるデータを識別し、ミラー化またはストライプ化を使用して、そのデータへのアクセス帯域幅を増やします。
- もっとも頻繁にアクセスされるデータを識別するには、使用可能なパフォーマンスモニタリング機能や、`iostat` コマンドなどの汎用のツールを使用します。識別されたら、ストライプ化、RAID-1 ボリューム、または RAID-5 ボリュームを使用して、このデータへのアクセス帯域幅を増やすことができます。
- ソフトパーティションのパフォーマンスは、ソフトパーティションのサイズが複数回変更されると低下する場合があります。
- RAID-5 ボリュームのパフォーマンスは、書き込み操作に対してストライプのパフォーマンスより低くなります。このパフォーマンスの低下は、RAID-5 ボリュームのパリティを計算して格納するために必要な複数の入出力操作からきています。

- 純粋なランダム入出力読み取りでは、ストライプと RAID-5 ボリュームは同等です。ストライプと RAID-5 ボリュームはどちらも、データを複数のディスクドライブにわたって分割します。RAID-5 ボリュームのパリティ計算は、スライス障害のあとを除き、読み取りでの要因にはなりません。
- 純粋なランダム入出力書き込みでは、ストライプは RAID-5 ボリュームより優れています。

Solaris Volume Manager に固有の構成ガイドラインについては、50 ページの「[Solaris Volume Manager 構成に関するガイドライン](#)」を参照してください。

ランダム入出力と順次入出力の最適化

このセクションでは、構成を最適化するための方法について説明します。

作成している Solaris Volume Manager ボリューム上で順次入出力とランダム入出力のどちらが多く発生するかがわかっていない場合は、これらのパフォーマンスチューニングのヒントを実装しないでください。これらのヒントが誤って実装されると、かえってパフォーマンス低下する場合があります。

次の最適化の提案では、RAID-0 ボリュームを最適化していることを前提にしています。一般には、RAID-0 ボリュームを最適化したあと、最適なパフォーマンスとデータの冗長性の両方を得るためにそのボリュームをミラー化することをお勧めします。

ランダム入出力

ランダム入出力環境(データベースや汎用のファイルサーバーに使用される環境など)では、すべてのディスクが入出力リクエストの処理に等しい時間を費やすようにしてください。

たとえば、データベースアプリケーションのための 40G バイトのストレージがあるとして、4つの 10G バイトディスクスピンドルにわたってストライプ化し、かつ入出力がランダムで、ボリューム間で均等に分散されている場合は、各ディスクが均一にビジー状態になるため、一般にはパフォーマンスが向上します。

`iostat` コマンドで報告されるように、ディスク上の最大のランダム入出力パフォーマンスの目標は 35% 以下の使用率です。通常、65% を超えるディスク使用率は問題になります。90% を超えるディスク使用率は重大な問題です。ディスク使用率の値が高すぎる場合の解決策は、より多くのディスク(スピンドル)を備えた新しい RAID-0 ボリュームを作成することです。

注-追加のディスクを既存のボリュームに単純に接続しても、パフォーマンスを向上させることはできません。パフォーマンスを最適化するための理想的なパラメータで新しいボリュームを作成する必要があります。

データをすべてのディスクにわたって分散させるだけであるため、ストライプのインタレースサイズは問題になりません。標準的な入出力リクエストを超えるインタレース値であれば十分です。

順次アクセス入出力

テーブル全体をスキャンすることの多いDBMSサーバーや、非常に大量のデータを処理する環境内のNFSサーバーなど、順次入出力環境内の構成のパフォーマンスを最適化できます。順次入出力環境を利用するには、インタレース値を標準的な入出力リクエストのサイズに対して相対的に低い値に設定します。

たとえば、標準的な入出力リクエストのサイズが256Kバイトであり、4つのスピンドルにわたってストライプ化されるとします。この例でのストライプユニットサイズの適切な選択は次のようになります。

$256\text{Kバイト} / 4 = 64\text{Kバイト}$ 、またはそれ以下

この方法により、標準的な入出力リクエストが複数のディスクスピンドルにわたって確実に分散されるようになるため、順次帯域幅が増加します。

注-順次入出力環境では、シーク時間とローテーション時間が実質的に0になります。順次入出力を最適化する場合は、ディスクの内部転送速度がもっとも重要になります。

順次アプリケーションでは通常、標準的な入出力サイズが大きく、128Kバイトを超える場合や、1Mバイトを超える場合もあります。あるアプリケーションの標準的な入出力リクエストのサイズが256Kバイトであり、4つのディスクスピンドルにわたってストライプ化されるとすると、次のようになります。

$256\text{Kバイト} / 4 = 64\text{Kバイト}$

そのため、インタレースサイズの適切な選択は32Kから64Kバイトになります。

Solaris Volume Manager の概要

この章では、Solaris Volume Manager の全体的な構造について説明します。この章では、次の内容について説明します。

- 39 ページの「Solaris Volume Manager の新機能」
- 39 ページの「Solaris Volume Manager の概要」
- 43 ページの「Solaris Volume Manager の要件」
- 43 ページの「Solaris Volume Manager コンポーネントの概要」
- 50 ページの「Solaris Volume Manager 構成に関するガイドライン」
- 51 ページの「Solaris Volume Manager コンポーネントの作成の概要」
- 52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」
- 53 ページの「Solaris Volume Manager へのアップグレード」

Solaris Volume Manager の新機能

このセクションでは、この Solaris リリースでの Solaris Volume Manager を使用する場合の新機能について説明します。

Solaris リリースの新機能のすべての一覧と説明については、『[Oracle Solaris 10 1/13 の新機能](#)』を参照してください。

Solaris Volume Manager の概要

Solaris Volume Manager は、多数のディスクやそれらのディスク上のデータを管理するために使用できるソフトウェア製品です。Solaris Volume Manager には多くの使用方法がありますが、ほとんどのタスクには次の利点があります。

- ストレージ容量の増加
- データ可用性の向上
- 大容量ストレージデバイスの管理の容易化

場合によっては、Solaris Volume Manager により入出力パフォーマンスも向上します。

Solaris オペレーティングシステムでサポートされるディスクタイプについては、『Oracle Solaris の管理: デバイスとファイルシステム』の第7章「ディスクの管理 (概要)」を参照してください。

Solaris Volume Manager でのストレージの管理方法

Solaris Volume Manager は仮想ディスクを使用して、物理ディスクとそれに関連するデータを管理します。Solaris Volume Manager では、仮想ディスクをボリュームと呼びます。また、歴史的な理由から、コマンド行ユーティリティーによってはボリュームをメタデバイスと呼ぶこともあります。

アプリケーションまたはファイルシステムから見た場合、ボリュームは機能的に物理ディスクと同じです。Solaris Volume Manager は、ボリューム宛ての入出力リクエストを、基となるメンバーディスクへの入出力リクエストに変換します。

Solaris Volume Manager ボリュームは、ディスクスライスまたはほかの Solaris Volume Manager ボリュームから構築されます。ボリュームを簡単に構築する方法として、Solaris 管理コンソールに組み込まれたグラフィカルユーザーインターフェース (GUI) の使用があります。Solaris 管理コンソール内の拡張ストレージツールは、ユーザーに既存のすべてのボリュームのビューを提供します。ウィザードの手順に従うことによって、任意の種類の Solaris Volume Manager ボリュームまたはコンポーネントを簡単に構築できます。また、Solaris Volume Manager コマンド行ユーティリティーを使用してボリュームを構築または変更することもできます。

たとえば、1つのボリュームとしてより多くのストレージ容量が必要な場合は、Solaris Volume Manager を使用して、システムでスライスの集まりを1つのより大きなボリュームとして処理するようにできます。これらのスライスからボリュームを作成したあと、ただちにそのボリュームを「実際の」スライスまたはデバイスの場合と同じように使用し始めることができます。

ボリュームのより詳細な説明については、[44 ページの「ボリュームの概要」](#)を参照してください。

Solaris Volume Manager は、RAID-1 (ミラー) ボリュームおよび RAID-5 ボリュームを使用して、データの信頼性と可用性を向上させることができます。Solaris Volume Manager ホットスペアは、ミラーや RAID-5 ボリュームに対して別のレベルのデータ可用性を提供できます。

構成を設定したら、Solaris 管理コンソール内の拡張ストレージツールを使用して、その操作に関する報告を行うことができます。

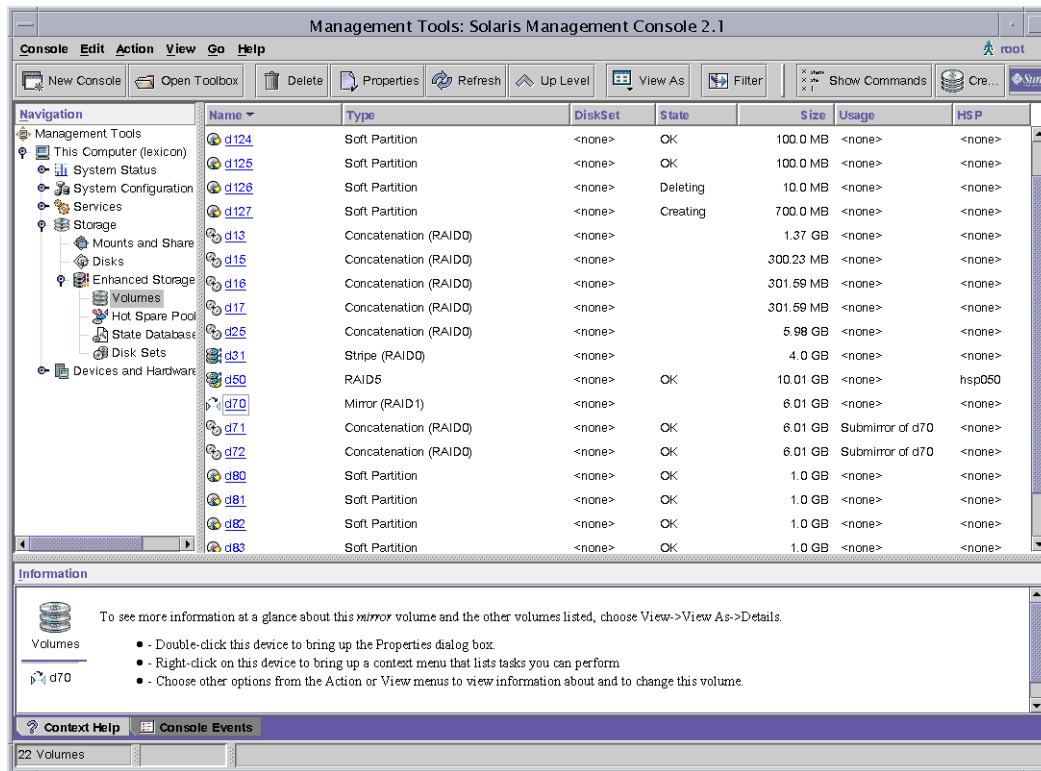
Solaris Volume Manager を管理する方法

Solaris Volume Manager を管理するには、次の方法のいずれかを使用します。

- Solaris 管理コンソール - このツールは、ボリューム管理機能を管理するための GUI を提供します。Solaris 管理コンソール内の拡張ストレージツールを使用します。拡張ストレージツールの例については、[図 3-1](#) を参照してください。このインタフェースは、ボリューム、ホットスペアプール、状態データベースの複製を含む、Solaris Volume Manager コンポーネントのグラフィカルビューを提供します。このインタフェースでは、Solaris Volume Manager コンポーネントのウィザードベースの操作が提供されるため、ディスクをすばやく構成したり、既存の構成を変更したりできます。
- コマンド行 - いくつかのコマンドを使用してボリューム管理機能を実行できます。Solaris Volume Manager のコアコマンドは、`meta` で始まります (`metainit` コマンドや `metastat` コマンドなど)。Solaris Volume Manager のコマンドの一覧については、[付録 B 「Solaris Volume Manager のクイックリファレンス」](#) を参照してください。

注 - Solaris Volume Manager をコマンド行と GUI で同時に管理しようとししないでください。構成に対して矛盾した変更が加えられる可能性があり、その動作が予測できなくなります。Solaris Volume Manager を管理するために両方のツールを使用できますが、同時には使用できません。

図 3-1 Solaris 管理コンソールでの拡張ストレージツール (Solaris Volume Manager) のビュー



▼ Solaris Volume Manager グラフィカルユーザーインターフェイス (GUI) にアクセスする方法

Solaris Volume Manager GUI (拡張ストレージ) は Solaris 管理コンソールの一部です。GUI にアクセスするには、次の手順を使用します。

- 1 次のコマンドを使用して、ホストシステムで Solaris 管理コンソールを起動します。
% /usr/sbin/smc
- 2 ナビゲーションペインで「このコンピュータ」をダブルクリックします。
- 3 ナビゲーションペインで「ストレージ」をダブルクリックします。
- 4 ナビゲーションペインで「拡張ストレージ」をダブルクリックして Solaris Volume Manager ツールをロードします。

- 5 ログインするよう求められたら、**root** または同等のアクセス権を持つユーザーとしてログインします。
- 6 ボリューム、ホットスペアプール、状態データベースの複製、およびディスクセットを管理するための適切なアイコンをダブルクリックします。

ヒント-Solaris 管理コンソールのすべてのツールでは、コンソールウィンドウの一番下のセクションか、またはウィザードパネルの左側に情報が表示されます。このインタフェースでのタスクについて情報が必要な場合は、いつでも「ヘルプ (Help)」を選択できます。

Solaris Volume Manager の要件

Solaris Volume Manager の要件には、次のものがあります。

- Solaris Volume Manager を管理するための **root** 権限を持っている必要があります。Solaris 管理コンソールのユーザープロファイル機能によって許可された同等の権限を使用すると、Solaris 管理コンソールからの管理が可能になります。ただし、Solaris Volume Manager のコマンド行インタフェースを使用できるのは **root** ユーザーだけです。
- Solaris Volume Manager でボリュームを作成するには、Solaris Volume Manager システム上に状態データベースの複製が存在している必要があります。状態データベースの複製には、すべてのボリューム、ホットスペア、ディスクセットの構成およびステータス情報が含まれています。信頼性を最大限に高めるために、少なくとも3つの複製が存在し、それらの複製が異なるコントローラおよび異なるディスク上に配置されるようにしてください。状態データベースの複製の詳細は、[67 ページの「Solaris Volume Manager 状態データベースと複製について」](#)を参照してください。状態データベースの複製を作成する方法については、[74 ページの「状態データベースの複製の作成」](#)

Solaris Volume Manager コンポーネントの概要

Solaris Volume Manager で作成するコンポーネントには、ボリューム、ソフトパーティション、ディスクセット、状態データベースの複製、ホットスペアプールの5つの基本型があります。次の表は、これらの Solaris Volume Manager 機能の概要を示しています。

表 3-1 Solaris Volume Manager 機能のサマリー

Solaris Volume Manager 機能	定義	目的	参照先
<ul style="list-style-type: none"> ■ RAID-0 ボリューム (ストライプ、連結、連結ストライプ) ■ RAID-1 (ミラー) ボリューム ■ RAID-5 ボリューム 	システムからは1つの論理デバイスとして認識される物理スライスのグループ。	ストレージ容量、パフォーマンス、またはデータ可用性を向上させるため。	44 ページの「ボリュームの概要」
ソフトパーティション	より小さく、より管理しやすいストレージユニットを提供するための、物理スライスまたは論理ボリュームの細分化。	大容量ストレージボリュームの管理容易性を向上させるため。	第 12 章「ソフトパーティション (概要)」
状態データベース (状態データベースの複製)	すべてのボリューム、ホットスペア、ディスクセットの構成およびステータス情報を含むデータベース。ユーザーが状態データベースの複製を作成するまで、Solaris Volume Manager は動作できません。	Solaris Volume Manager 構成の状態に関する情報を格納するため。	48 ページの「状態データベースと状態データベースの複製」
ホットスペアプール	予約されたスライス (ホットスペア) の集まり。サブミラーまたは RAID-5 ボリュームのどちらかのコンポーネントに障害が発生すると、これらのスライスは自動的に置き換えられます。	RAID-1 および RAID-5 ボリュームのデータ可用性を向上させるため。	49 ページの「ホットスペアプール」
ディスクセット	ボリュームとホットスペアを含み、複数のホストが同時にではなく共有できる、個別の名前空間内の一連の共有ディスクドライブ。	データの冗長性とデータ可用性を実現するとともに、個別の名前空間を提供して管理を容易にするため。	49 ページの「ディスクセット」

ボリュームの概要

ボリュームは、システムからは1つの論理デバイスとして認識される物理スライスのグループです。ボリュームは、一般的な UNIX 用語である「擬似 (仮想) デバイス」と、実質的に同義です。

注 - 従来より、Solstice DiskSuite 製品ではこれらの論理デバイスをメタデバイスと呼んでいます。ただし、簡単にするため、また標準化のために、このドキュメントではこれらのデバイスをボリュームと呼びます。

ボリュームのクラス

ボリュームは、RAID-0(連結またはストライプ)ボリューム、RAID-1(ミラー)ボリューム、RAID-5ボリューム、またはソフトパーティションとして作成します。

Solaris 管理コンソール内の拡張ストレージツールまたはコマンド行ユーティリティーのどちらかを使用して、ボリュームを作成および管理できます。

次の表は、ボリュームのクラスの要約を示しています。

表 3-2 ボリュームのクラス

ボリューム	説明
RAID-0(ストライプまたは連結)	直接使用するか、またはミラーの基本構成ブロックとして使用できません。RAID-0 ボリュームでは、データの冗長性が直接には提供されません。
RAID-1(ミラー)	複数のコピーを保持することによってデータを複製します。RAID-1 ボリュームは、サブミラーと呼ばれる 1 つまたは複数の RAID-0 ボリュームから構成されます。
RAID-5	パリティ情報を使用してデータを複製します。ディスク障害が発生した場合は、使用可能なデータとパリティ情報を使用して、欠落したデータを再生成できます。RAID-5 ボリュームは一般に、スライスで構成されませんが、そのパリティは RAID-5 ボリューム内のすべてのスライスにわたって分散されます。
ソフトパーティション	スライスまたは論理ボリュームを 1 つまたは複数の拡張可能な、より小さなボリュームに分割します。

ボリュームの使用方法

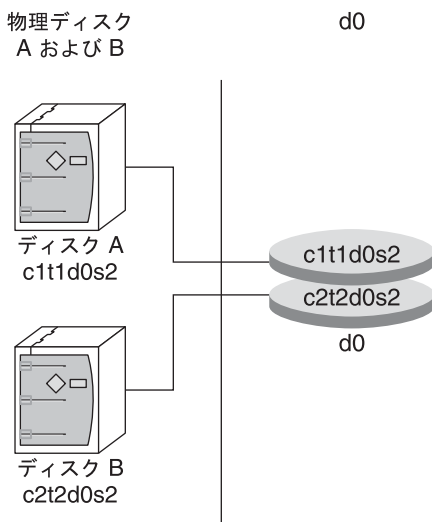
ボリュームは、ストレージ容量、パフォーマンス、およびデータ可用性を向上させるために使用します。場合によっては、ボリュームの使用により入出力パフォーマンスも向上します。ボリュームの機能は、スライスと同じです。ボリュームはスライスとして見えるため、ボリュームはエンドユーザー、アプリケーション、およびファイルシステムに対して透過的です。物理デバイスと同様に、ボリュームはブロックデバイス名または raw デバイス名でアクセスされます。ボリューム名は、使用しているのがブロックデバイスなのか raw デバイスなのかによって異なります。ボリューム名の詳細は、[47 ページの「ボリューム名」](#)を参照してください。

ボリューム上では、mkfs、mount、umount、ufsdump、ufsrestoreなどを含むほとんどのファイルシステムコマンドを使用できます。ただし、formatコマンドは使用できません。ボリュームにマウントされたファイルシステムが含まれているかぎり、そのボリュームとの間でファイルの読み取り、書き込み、およびコピーを行うことができます。

例-2つのスライスで構成されたボリューム

図3-2は、ディスクAからの1つのスライスと、ディスクBからの1つのスライスの2つのスライスを含むボリュームを示しています。アプリケーションやUFSは、このボリュームを1つの物理ディスクであるかのように処理します。このボリュームにさらにスライスを追加すると、そのストレージ容量が増加します。

図3-2 ボリューム、物理ディスク、およびスライスの関係



growfs コマンドを使用したボリュームおよびディスク容量の拡張

Solaris Volume Manager では、スライスを追加することによってボリュームを拡張できます。Solaris 管理コンソール内の拡張ストレージツールまたはコマンド行インタフェースのどちらかを使用して、既存のボリュームにスライスを追加できます。

ボリューム内に含まれているマウントされた、またはマウントされていない UFS ファイルシステムを、システムの停止やバックアップを行うことなく拡張できます。ただし、データは常にバックアップすることをお勧めします。ボリュームを拡張したあと、growfs コマンドを使用してファイルシステムを拡張します。

注-ファイルシステムが拡張されたあとに、そのファイルシステムのサイズを減らすことはできません。ファイルシステムのサイズを減らすことができない点はUFSの制限です。同様に、Solaris Volume Manager パーティションのサイズを増やしたあとに、そのサイズを減らすことはできません。

raw ボリュームを使用するアプリケーションやデータベースは、追加された容量を「拡張」してアプリケーションが認識できるようにするための独自の方法を備えている必要があります。Solaris Volume Manager には、この機能は用意されていません。

ボリューム内のディスク容量は、次の方法で拡張できます。

- RAID-0 ボリュームへの1つまたは複数のスライスの追加
- RAID-1 ボリュームのすべてのサブミラーへの1つまたは複数のスライスの追加
- RAID-5 ボリュームへの1つまたは複数のスライスの追加
- 基となるコンポーネントから容量を追加することによるソフトパーティションの拡張

growfs コマンドは、サービスやデータを失うことなくUFSファイルシステムを拡張します。ただし、growfs コマンドの実行中、そのボリュームへの書き込みアクセスは中断されます。ファイルシステムは、そのファイルシステムを含むスライスまたはボリュームのサイズまで拡張できます。

growfs コマンドの `-s size` オプションを使用すると、追加のディスク容量の一部のみを使用するようにファイルシステムを拡張できます。

注-ミラーを拡張すると、そのミラーの基となるサブミラーに容量が追加されます。次に、growfs コマンドがRAID-1 ボリューム上で実行されます。一般的な規則として、基となるデバイスに容量が追加されたあと、growfs コマンドがトップレベルのデバイス上で実行されます。

ボリューム名

物理スライスと同様に、ボリュームには、ファイルシステムで表示される論理名があります。論理ボリューム名は、ブロックデバイスの場合は `/dev/md/dsk` ディレクトリ、raw デバイスの場合は `/dev/md/rdisk` ディレクトリ内にエントリがあります。ボリューム名の全体 (`/dev/md/dsk/volume-name` など) を指定する代わりに、多くの場合は、どの `meta*` コマンドでも省略されたボリューム名 (`d1` など) を使用できます。一般に、ボリュームが現在使用されておらず、新しい名前が別のボリュームによって使用されていないかぎりには、ボリュームの名前を変更できます。詳細は、[241 ページ](#)の「**ボリューム名の交換**」を参照してください。

当初、ボリューム名は文字「d」で始まり、そのあとに1つの数字が続く必要がありました (`d0` など)。この形式は引き続き使用できます。「`d*`」の命名構造を使用するボリューム名の例を次に示します。

/dev/md/dsk/d0	ブロックボリューム d0
/dev/md/dsk/d1	ブロックボリューム d1
/dev/md/rdisk/d126	raw ボリューム d126
/dev/md/rdisk/d127	raw ボリューム d127

ボリューム名に関するガイドライン

ボリューム名に関する標準を使用することによって、管理が単純化され、ボリュームタイプをひと目で識別できるようになります。次に、いくつかの提案を示します。

- ボリュームのタイプごとに範囲を使用します。たとえば、RAID-1 ボリュームには 0 から 20、RAID-0 ボリュームには 21 から 40 の数字を割り当て、以降も同様とします。
- ミラーには名前の関係を使用します。たとえば、ミラーには 0 で終わる数字を使用した名前を、サブミラーには 1、2 など で終わる数字を使用した名前を付けます。たとえば、ミラー d10、サブミラー d11 および d12 や、ミラー d20、サブミラー d21、d22、d23、および d24 などの名前を付けることができます。
- スライス番号とディスク番号がボリューム番号に対応するような命名方法を使用します。

状態データベースと状態データベースの複製

状態データベースは、Solaris Volume Manager 構成の状態に関する情報を格納するデータベースです。状態データベースは、構成に加えられた変更を記録および追跡します。Solaris Volume Manager は、構成または状態の変更が発生すると、状態データベースを自動的に更新します。新しいボリュームの作成は、構成の変更の一例です。状態の変更の 1 つの例として、サブミラーの障害があります。

状態データベースは、実際には、複製された複数のデータベースコピーの集まりです。各コピーは、「状態データベースの複製」と呼ばれ、データベース内のデータが常に有効であることを保証します。状態データベースの複数のコピーを保持することにより、シングルポイント障害によるデータ損失から保護されます。状態データベースは、既知のすべての状態データベースの複製の場所とステータスを追跡します。

ユーザーが状態データベースとその状態データベースの複製を作成するまで、Solaris Volume Manager は動作できません。Solaris Volume Manager 構成には、正常に機能している状態データベースが必要です。

構成を設定する場合は、状態データベースの複製を次の場所のどちらかに格納できません。

- 専用スライス上
- あとでボリュームの一部になるスライス上

Solaris Volume Manager は、スライスに状態データベースの複製がいつ含まれているかを認識し、そのスライスがボリュームで使用されている場合はその複製を自動的にスキップします。状態データベースの複製のために予約されているスライスの部分を、ほかのどの目的にも使用してはいけません。

複数の状態データベースのコピーを1つのスライス上に置くこともできます。ただし、そうすることにより、システムがシングルポイント障害に対してより脆弱になる可能性があります。

状態データベースの複製がすべて削除された場合でも、Solaris オペレーティングシステムは引き続き正常に機能します。ただし、ディスク上に既存の状態データベースの複製がない状態でリポートが発生した場合、システムはすべての Solaris Volume Manager 構成データを失います。

ホットスペアプール

ホットスペアプールは、障害が発生したコンポーネントのために自動的に置き換えられるように Solaris Volume Manager によって予約されたスライス (ホットスペア) の集まりです。これらのホットスペアは、サブミラーまたは RAID-5 ボリュームのどちらかで使用できます。ホットスペアによって、RAID-1 および RAID-5 ボリュームのデータ可用性が向上します。Solaris 管理コンソール内の拡張ストレージツールまたはコマンド行インタフェースのどちらかを使用して、ホットスペアプールを作成できます。

コンポーネントのエラーが発生した場合、Solaris Volume Manager は、障害が発生したコンポーネントのサイズ以上のサイズを持つ使用可能な最初のホットスペアをチェックします。見つかった場合、Solaris Volume Manager はそのコンポーネントを自動的に置き換え、データを再同期します。ホットスペアのリストに十分なサイズのスライスが見つからなかった場合は、サブミラーまたは RAID-5 ボリュームに障害が失敗したと見なされます。詳細は、[第 16 章「ホットスペアプール \(概要\)」](#)を参照してください。

ディスクセット

ディスクセットは論理ボリュームとホットスペアを含む一連の物理ストレージボリュームです。ボリュームとホットスペアプールは、そのディスクセット内からの

ドライブ上に構築する必要があります。ディスクセット内にボリュームを作成したら、そのボリュームを物理スライスの場合と同様に使用できます。

ディスクセットによって、クラスタ化された環境内のデータ可用性が提供されます。1つのホストに障害が発生した場合は、その障害が発生したホストのディスクセットを別のホストが引き継ぐことができます。(このタイプの構成は、フェイルオーバー構成と呼ばれます。)さらに、ディスクセットを使用して Solaris Volume Manager 名前空間の管理に役立てたり、ネットワーク接続ストレージデバイスへのすぐ可以使用できるアクセスを提供したりできます。

詳細は、第18章「ディスクセット(概要)」を参照してください。

Solaris Volume Manager 構成に関するガイドライン

Solaris Volume Manager 構成の不適切な設計によって、パフォーマンスが低下する場合があります。このセクションでは、Solaris Volume Manager から高いパフォーマンスを引き出すためのヒントを提供します。ストレージ構成のパフォーマンスに関するガイドラインについては、35 ページの「一般的なパフォーマンスに関するガイドライン」を参照してください。

一般的なガイドライン

- ディスクおよびコントローラ-ドライブを独立したドライブパス上のボリューム、または SCSI ドライブの場合は独立したホストアダプタ上のボリュームに配置してください。入出力負荷を複数のコントローラにわたって分散させると、ボリュームのパフォーマンスと可用性が向上します。
- システムファイル `/etc/lvm/mddb.cf` または `/etc/lvm/md.cf` ファイルを決して編集したり削除したりしないでください。
これらのファイルが定期的にバックアップされることを確認してください。
- ボリュームの完全性-スライスがボリュームとして定義されている場合は、ダンプレデバイスとしてのスライスの使用も含め、基となるスライスをほかのどの目的にも使用しないでください。
- ディスクとパーティションに関する情報-不良ディスクの再フォーマットや Solaris Volume Manager 構成の再作成が必要になった場合に備えて、`prtvtoc` および `metastat -p` コマンドからの出力のコピーを保管しておいてください。

ファイルシステムに関するガイドライン

ファイルシステムをボリュームの基となるスライスにマウントしないでください。スライスが何らかの種類のボリュームのために使用されている場合は、そのスライスをファイルシステムとしてマウントしてはいけません。可能な場合は、ボリュームをアクティブにする前に、そのボリュームとして使用しようとしている物理デバイスをすべてアンマウントしてください。

Solaris Volume Manager コンポーネントの作成の概要

Solaris Volume Manager コンポーネントを作成する場合は、論理的な Solaris Volume Manager の名前 (`d0` など) に物理スライスを割り当てます。作成できる Solaris Volume Manager コンポーネントには、次のものがあります。

- 状態データベースの複製
- ボリューム (RAID-0 (ストライプ、連結)、RAID-1 (ミラー)、RAID-5、およびソフトパーティション)
- ホットスペアプール
- ディスクセット

注- ボリュームの命名方法に関する提案については、[47 ページ](#)の「[ボリューム名](#)」を参照してください。

Solaris Volume Manager コンポーネントを作成するための前提条件

Solaris Volume Manager コンポーネントを作成するための前提条件は次のとおりです。

- 初期の状態データベースの複製を作成します。まだ作成していない場合は、[74 ページ](#)の「[状態データベースの複製の作成](#)」を参照してください。
- Solaris Volume Manager で使用できるスライスを識別します。必要に応じて、`format` コマンド、`fmthard` コマンド、または Solaris 管理コンソールを使用して既存のディスクをパーティション分割し直します。
- `root` 権限があることを確認します。
- すべてのデータの現在のバックアップを作成します。

- GUIを使用している場合は、Solaris 管理コンソールを起動し、Solaris Volume Manager の機能に移動します。詳細は、42 ページの「Solaris Volume Manager グラフィカルユーザーインタフェース (GUI) にアクセスする方法」を参照してください。

Solaris Volume Manager でのマルチテラバイトのサポートの概要

Solaris 9 4/03 以降のリリースでは、Solaris Volume Manager は、64 ビットカーネルを実行しているシステム上で 1T バイトを超えるストレージデバイスおよび論理ボリュームをサポートしています。

注 - システムが 64 ビットカーネルを実行しているかどうかを確認するには、`isainfo -v` を使用します。文字列「64-bit」が表示された場合は、64 ビットカーネルが実行されています。

Solaris Volume Manager では、次のことを実行できます。

- 1T バイトを超えるサイズの論理ストレージユニット (LUN) 上に、またはこれらの LUN から構築された論理ボリュームを作成、変更、および削除します。
- 1T バイトを超えるサイズの論理ボリュームを作成、変更、および削除します。

大容量ボリュームのサポートは自動的に行われます。1T バイトを超えるデバイスが作成された場合、ユーザーの介入がなくても、Solaris Volume Manager はそれを適切に構成します。

大容量ボリュームのサポートの制限

Solaris Volume Manager は、64 ビットカーネルを実行している Solaris 9 4/03 以降のリリースでのみ (1T バイトを超える) 大容量ボリュームをサポートします。大容量ボリュームを含むシステムを、以前の Solaris 9 リリースの 32 ビットカーネルで実行すると、Solaris Volume Manager の機能に影響を与えます。具体的には、次の点に注意してください。

- 大容量ボリュームを含むシステムが 32 ビットの Solaris 9 4/03 以降のカーネルでリブートされた場合、大容量ボリュームは `metastat` の出力を通して表示されますが、アクセス、変更、または削除はできません。さらに、新しい大容量ボリュームの作成もできません。大容量ボリューム上のボリュームまたはファイルシステムもすべて使用できません。

- 大容量ボリュームを含むシステムが Solaris 9 4/03 より前の Solaris リリースでリブートされた場合、Solaris Volume Manager は起動しません。Solaris Volume Manager を別のバージョンの Solaris プラットフォームで実行するには、すべての大容量ボリュームを削除する必要があります。



注意 - 32 ビットカーネルで Solaris ソフトウェアを実行することを予定している場合や、Solaris 9 4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、大容量ボリュームを作成しないでください。

大容量ボリュームの使用

Solaris Volume Manager コマンドはすべて、大容量ボリュームで動作します。大容量ボリュームのサポートを利用するために、構文の違いや特殊なタスクは必要ありません。そのため、Solaris Volume Manager に精通しているシステム管理者は、ただちに Solaris Volume Manager の大容量ボリュームを操作できます。

ヒント - 大容量ボリュームを作成したあとで、以前の Solaris リリースで Solaris Volume Manager を使用する必要があるか、または 32 ビットの Solaris 9 4/03 以降のカーネルで実行する必要があることがわかった場合は、それらの大容量ボリュームを削除する必要があります。以前の Solaris リリースまたは 32 ビットカーネルでリブートする前に、64 ビットカーネルで `metaclear` コマンドを使用して Solaris Volume Manager 構成から大容量ボリュームを削除してください。

Solaris Volume Manager へのアップグレード

Solaris Volume Manager は、Solstice DiskSuite バージョン 4.1、4.2、および 4.2.1 からのシームレスなアップグレードを完全にサポートします。すべてのボリュームが（「保守が必要」や「最後にエラー」ではなく）「正常」状態にあること、および使用中のホットスペアがないことを確認してください。アップグレードを機能させるために、Solaris Volume Manager に対してほかに特別なことを行う必要はありません。構成を変更したり、ルートミラーを解除したりする必要はありません。システムをアップグレードすると、Solstice DiskSuite 構成が引き継がれ、Solaris Volume Manager ツールによるアップグレードのあとにアクセス可能になります。

Solaris 10 OS では、従来の UNIX の起動スクリプト、`init` 実行レベル、および構成ファイルを補強するためのインフラストラクチャーを提供するサービス管理機能 (SMF) が導入されました。以前のバージョンの Solaris OS からアップグレードしている場合は、Solaris Volume Manager に関連付けられた SMF サービスがオンラインであることを確認してください。SMF サービスがオンラインでない場合は、Solaris Volume Manager の管理に関する問題が発生することがあります。

Solaris Volume Manager に関連付けられた SMF サービスを確認するには、次の形式の `svcs` コマンドを使用します。

```
# svcs -a |egrep "md|meta"
disabled      12:05:45 svc:/network/rpc/mdcomm:default
disabled      12:05:45 svc:/network/rpc/metamed:default
disabled      12:05:45 svc:/network/rpc/metamh:default
online        12:05:39 svc:/system/metainit:default
online        12:05:46 svc:/network/rpc/meta:default
online        12:05:48 svc:/system/fmd:default
online        12:05:51 svc:/system/mdmonitor:default
```

Solaris Volume Manager 構成がローカルセットのみで構成されている場合は、次のサービスをオンラインにしてください。

```
svc:/system/metainit
svc:/network/rpc/meta
svc:/system/mdmonitor
```

Solaris Volume Manager 構成にディスクセットが含まれている場合は、次の追加のサービスをオンラインにしてください。

```
svc:/network/rpc/metamed
svc:/network/rpc/metamh
```

Solaris Volume Manager にマルチノードのディスクセットが含まれている場合は、すでに挙げたほかのサービスに加えて、次のサービスをオンラインにしてください。

```
svc:/network/rpc/mdcomm
```

SMF の詳細は、『[Oracle Solaris の管理: 基本管理](#)』の第 18 章「サービスの管理 (概要)」を参照してください。

◆ ◆ ◆ 第 4 章

Solaris Volume Manager for Sun Cluster (概要)

この章では、Solaris Volume Manager for Sun Cluster の概要について説明します。

この章の内容は次のとおりです。

- 55 ページの「Solaris Volume Manager for Sun Cluster の概要」
- 57 ページの「複数所有者ディスクセットの概念」
- 60 ページの「Solaris Volume Manager for Sun Cluster 構成」
- 60 ページの「複数所有者ディスクセット内の RAID-1 (ミラー) ボリューム」

Solaris Volume Manager for Sun Cluster の概要

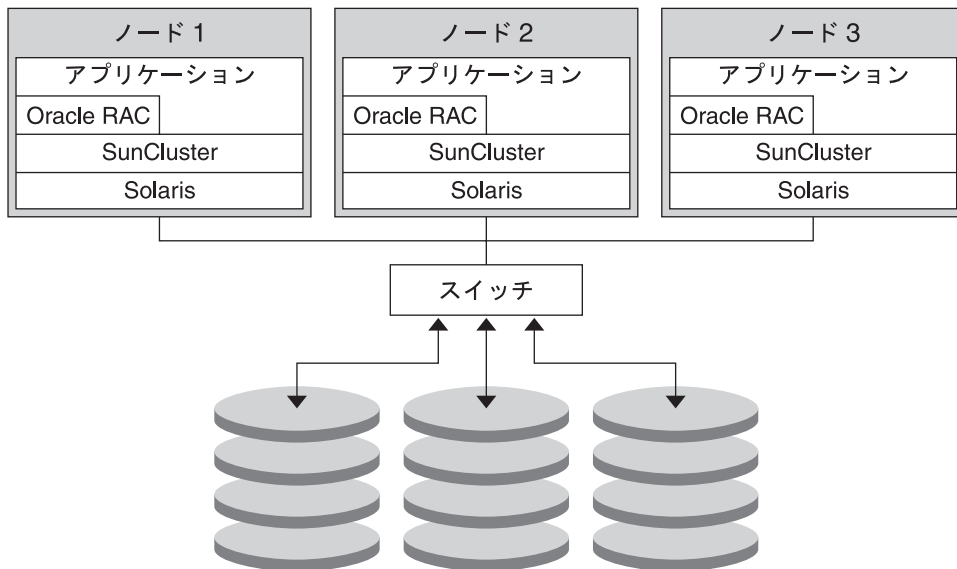
Solaris 9 9/04 以降のリリースでは、Solaris Volume Manager は、複数所有者ディスクセットを使用して Sun Cluster 環境内のストレージを管理できます。「複数所有者ディスクセット」を使用すると、複数のノードがディスクセットの所有権を共有したり、共有ディスクに同時に書き込んだりすることができます。以前は、共有ディスクセットはそのディスクセットに参加しているすべてのホストから表示できましたが、一度にアクセスできるのは1つのホストだけでした。複数所有者ディスクセットは、Sun Cluster や、Oracle Real Application Clusters などのアプリケーションで動作します。

複数所有者ディスクセットと Solaris Volume Manager 共有ディスクセットは、同じノード上に共存できます。ただし、2つの構成の間でのディスクセットの移動はサポートされていません。

Solaris Volume Manager for Sun Cluster は、ストライプ、連結、ミラー、ソフトパーティション、ホットスペアなどの、Solaris Volume Manager で作成できるのと同じコンポーネントを作成します。Solaris Volume Manager for Sun Cluster は、RAID-5 ボリュームやトランザクションボリュームはサポートしていません。

次の図は、標準的なクラスタ構成でのソフトウェアと共有ストレージの間の関連付けを示しています。

図4-1 サンプルのクラスタ構成



各ノードには、ローカルストレージのほかに、共有ストレージへの少なくとも1つのバスがあります。クラスタ内の複数所有者ディスクセットは、Solarisオペレーティングシステム (Solaris OS) の一部である Solaris Volume Manager for Sun Cluster によって管理されます。

前提条件: 複数所有者ディスクセット機能に必要なソフトウェアコンポーネント

Solaris Volume Manager for Sun Cluster を使用するには、Solaris OSに加えて、次のソフトウェアがインストールされている必要があります。

- Sun Cluster 初期クラスタフレームワーク
- Sun Cluster Support for Oracle Real Application Clusters ソフトウェア
- Oracle Real Application Clusters ソフトウェア

注 - Sun Cluster および Oracle Real Application Clusters ソフトウェアの設定については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』と『[Oracle Solaris Cluster Data Service for Oracle Real Application Clusters ガイド](#)』を参照してください。

複数所有者ディスクセットの概念

Solaris Volume Manager for Sun Cluster によって管理されるストレージは、複数所有者ディスクセットにグループ化されます。複数所有者ディスクセットを使用すると、複数のノードがディスクセットの所有権を共有したり、共有ディスクに同時に書き込んだりすることができます。Oracle Real Application Clusters などのアプリケーションのインスタンスはクラスタ内の各ノード上で実行されるため、複数所有者ディスクセットによってスケラビリティが提供されます。アプリケーションの各インスタンスは共有ストレージに直接アクセスするため、複数所有者ディスクセットによって、そのアプリケーションのパフォーマンスも向上します。

注 - 複数所有者ディスクセット機能は、Sun Cluster 環境でのみ有効になります。「ノード」は、Sun Cluster システムの一部である物理マシンです。

各複数所有者ディスクセットは、ノードのリストに関連付けられています。これらのノードが、そのディスクセットの所有権を共有します。次の `metaset -s disk-set` コマンドは、複数所有者ディスクセットの出力を表示します。

```
# metaset -s blue
```

```
Multi-owner Set name = blue, Set number = 1, Master = nodeone
```

Host	Owner	Member
nodeone	multi-owner	Yes
nodetwo	multi-owner	Yes

```
Drive Dbase
```

```
d9 Yes
```

```
d13 Yes
```

この出力は、ディスクセットの所有権を共有する、ノードのリスト内の `nodeone` と `nodetwo` を示しています。さらに、`nodeone` は「マスターノード」として指定されています。

各複数所有者ディスクセットにはマスターノードがあります。ディスクセットが作成されたあと、そのディスクセットに最初のディスクを追加したノードがディスクセットのマスターノードになります。マスターノードは、そのディスクセット内の状態データベースの複製を作成、削除、および更新します。

注 - 状態データベースの複製の詳細は、[第6章「状態データベース\(概要\)」](#)を参照してください。

Solaris Volume Manager for Sun Cluster は、ノードリストが異なっているが、重複しているディスクセットをサポートできます。各ディスクセットにはマスターノードがあるため、同じクラスタ上に複数のマスターが同時に存在できます。

metaset コマンドの次の出力は、ディスクセットに最初のディスクが追加されたときに nodeone がマスターノードになることを示しています。

```
nodeone# metaset -s red
Multi-owner Set name = red, Set number = 1, Master =

Host          Owner          Member
nodeone      nodeone        Yes
nodetwo      nodeone        Yes
nodeone# metaset -s red -a /dev/did/dsk/d9
nodeone# metaset -s red

Multi-owner Set name = red, Set number = 1, Master = nodeone

Host          Owner          Member
nodeone      multi-owner    Yes
nodetwo      multi-owner    Yes

Drive    Dbase
d9       Yes
```

Solaris Volume Manager for Sun Cluster は、ノードリストが異なっているが、重複しているディスクセットをサポートできます。各ディスクセットにはマスターノードがあるため、同じクラスタ上に複数のマスターが同時に存在できます。

複数所有者ディスクセットに関連したタスク



注意 - 複数所有者ディスクセットを構成する前に、Solaris OS に加えて、次のソフトウェアがインストールされている必要があります。

- Sun Cluster 初期クラスタフレームワーク
- Sun Cluster Support for Oracle Real Application Clusters ソフトウェア
- Oracle Real Application Clusters ソフトウェア

Sun Cluster および Oracle Real Application Clusters ソフトウェアの設定については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』と『[Oracle Solaris Cluster Data Service for Oracle Real Application Clusters ガイド](#)』を参照してください。

Solaris Volume Manager for Sun Cluster は一般に、ディスクセットに関連したタスクを実行するために、同じ一連の Solaris Volume Manager コマンドを使用します。metaset コマンドには、複数所有者ディスクセットに固有のいくつかのコマンドオプションが追加されました。たとえば、複数所有者ディスクセットを作成するタスクに

は、`metaset` コマンドに `-M` が必要です。次の出力は、`metaset -s diskset-name -a -M -h hostname` コマンドを使用して複数所有者ディスクセットを作成する方法を示しています。

```
# metaset -s red -a -M -h nodeone
# metaset
Multi-owner Set name = red, Set number = 1, Master =

Host          Owner          Member
nodeone                               Yes
```

さらに、一部の `metaset` コマンドオプション (ディスクセットを取得および解放するコマンドなど) は、複数所有者ディスクセットでは使用されません。詳細は、[metaset\(1M\)](#) のマニュアルページを参照してください。

Sun Cluster 環境内のタスクがどのように異なるかを示す別の例が、ディスクの操作時に発生します。Sun Cluster は、各ディスクに一意的なデバイス ID (DID) 番号を割り当てます。ディスクを識別するために `cnt ndn` 形式を使用するのではなく、Sun Cluster の DID パス名 `/dev/did/dsk/dN` を使用します。変数 `N` は、Sun Cluster によって割り当てられたデバイス番号です。

次の出力は、`metaset -s diskset-name -a disk-name` コマンドと、ディスクを識別するための Sun Cluster の DID パス名を使用して、複数所有者ディスクセットにディスクを追加する方法を示しています。

```
nodeone# metaset -s red
Multi-owner Set name = red
Multi-owner Set name = red, Set number = 1, Master =

Host          Owner          Member
nodeone                               Yes
nodetwo                               Yes
nodeone# metaset -s red -a /dev/did/dsk/d13
nodeone# metaset -s red
Multi-owner Set name = red, Set number = 1, Master = nodeone

Host          Owner          Member
nodeone       multi-owner  Yes

Drive Dbase

d13  Yes
```

Oracle Real Application Clusters での複数所有者ディスクセットの作成については、『[Oracle Solaris Cluster Data Service for Oracle Real Application Clusters ガイド](#)』を参照してください。

ディスクセットに関連したタスクについては、[第 19 章「ディスクセット \(タスク\)」](#) を参照してください。

Solaris Volume Manager for Sun Cluster 構成

Solaris Volume Manager for Sun Cluster は、次の構成をサポートしています。

- Solaris Volume Manager for Sun Cluster は、最大 32 のディスクセットをサポートしています。これらのディスクセットには、複数所有者ディスクセット、共有ディスクセット、およびローカルディスクセットの任意の組み合わせを含めることができます。

注- 各種のタイプのディスクセットの詳細は、[206 ページの「ディスクセットのタイプ」](#)を参照してください。

- 各複数所有者ディスクセットは、ディスクセットごとに最大 8192 のボリュームをサポートしています。
- 状態データベースの複製のデフォルトのサイズは 16M バイトです。最小サイズは 16M バイトです。最大サイズは 256M バイトです。

再構成プロセス内の手順では、Sun Cluster Support for Oracle Real Application Clusters の多くの拡張プロパティがタイムアウトを指定します。タイムアウトの設定の詳細は、『[Oracle Solaris Cluster Data Service for Oracle Real Application Clusters ガイド](#)』を参照してください。

複数所有者ディスクセット内の RAID-1 (ミラー) ボリューム

複数所有者ディスクセット内に作成された RAID-1 ボリューム (またはミラー) は、Solaris Volume Manager 共有ディスクセット内の RAID-1 ボリュームと同様に機能します。ただし、複数所有者ディスクセット内の RAID-1 ボリュームには、いくつかの追加機能があります。

複数所有者ディスクセットのミラー所有権

ミラー所有権は、複数所有者ディスクセットに固有の概念です。Solaris Volume Manager 共有ディスクセット内の RAID-1 ボリュームとは異なり、複数所有者ディスクセット内の RAID-1 ボリュームには通常、所有者が関連付けられています。ミラーボリュームの所有権は、ボリュームマネージャーによって選択されます。ボリュームの所有者は、ディスクセットのノードリストで指定されているノードのいずれかです。RAID-1 ボリュームの所有者だけがそのボリュームに書き込むことができます。所有者以外のノードがボリュームに書き込もうとした場合、所有権は、書

き込み操作を行なっているノードに切り替わります。metastat -s diskset-name コマンドの次の出力は、RAID-1 ボリューム d24 の所有者としての nodeone を示しています。

```
# metastat -s red
red/d24: Mirror
  Submirror 0: red/d20
    State: Okay
  Submirror 1: red/d21
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Resync option: optimizedresync
  Owner: nodeone
  Size: 825930 blocks (403 MB)
```

データの管理と回復のプロセス

Solaris Volume Manager の RAID-1 ボリュームと同様に、Solaris Volume Manager for Sun Cluster の RAID-1 ボリュームでは、一貫性のあるデータを保証するための操作が実行されます。Solaris Volume Manager for Sun Cluster の RAID-1 ボリュームには、データの管理と回復のための2つのオプションが用意されています。

Solaris Volume Manager for Sun Cluster での最適化された再同期

Solaris Volume Manager for Sun Cluster での最適化された再同期は、Solaris Volume Manager での最適化された再同期と同様に機能します。ただし、複数所有者ディスクセットでは、再同期オプションが最適化された再同期に設定された RAID-1 ボリュームには、常にミラー所有者が関連付けられています。metastat -s diskset-name コマンドの次の出力は、optimizedresync (最適化された再同期) に設定された再同期オプションを示しています。

```
# metastat -s red
red/d24: Mirror
  Submirror 0: red/d20
    State: Okay
  Submirror 1: red/d21
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Resync option: optimizedresync
  Owner: nodeone
  Size: 825930 blocks (403 MB)
```

最適化された再同期の詳細は、103 ページの「最適化された再同期」を参照してください。

アプリケーションベースの回復と指定されたミラー読み取り

Solaris Volume Manager for Sun Cluster でのデータ回復を最適化するために、Oracle Real Application Clusters などのアプリケーションには、データの回復を管理および制御するための機能が必要になります。アプリケーションが回復を制御できるようにすると、回復のパフォーマンスが向上します。

DKIOGETVOLCAP、DKIOSETVOLCAP、DKIODMR の各 ioctl は、クラスタ環境でのアプリケーションのデータ管理回復に対するサポートを提供します。これらの ioctl は、アプリケーションに次の機能を提供します。

- アプリケーションベースの回復 (ABR) - アプリケーションがミラー化ボリューム上のデータの回復を制御できるようにします。
- 指定されたミラー読み取り - アプリケーションが読み取りを特定のサブミラーに指定したり、データの状態を確認したりできるようにします。

アプリケーションベースのデータ管理回復で使用される ioctl の詳細は、[dkio\(7I\)](#) のマニュアルページを参照してください。

再同期オプションがアプリケーションベースの回復に設定された RAID-1 ボリュームには、アプリケーションベースの回復プロセス中にのみミラー所有者が関連付けられます。metastat -s diskset-name コマンドの次の出力は、正常な状態にある RAID-1 ボリュームを示しています。再同期オプションは、アプリケーションベースの回復に設定されています。ミラー所有者は存在しません。

```
# metastat -s red
red/d24: Mirror
  Submirror 0: red/d20
    State: Okay
  Submirror 1: red/d21
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Resync option: application based
  Owner: None
  Size: 825930 blocks (403 MB)
```

Solaris Volume Manager の構成と使用 (シナリオ)

『Solaris Volume Manager 管理ガイド』の全体を通して、例は一般に (可能な場合は常に) 1 つのストレージ構成に関連しています。この章では、これらの例で使用されているシナリオについて説明します。この章では、以降の章で使用されている初期のストレージ構成に関する詳細について説明します。

この章では、次の内容について説明します。

- 63 ページの「シナリオの背景情報」
- 65 ページの「最終的な Solaris Volume Manager 構成」

シナリオの背景情報

このドキュメントの全体を通して、各シナリオや、例の多くは 1 つの構成に関連しています。この構成は (ドキュメントを単純化するために) 小規模ですが、その概念は、はるかに大規模なストレージ環境にも拡張されます。

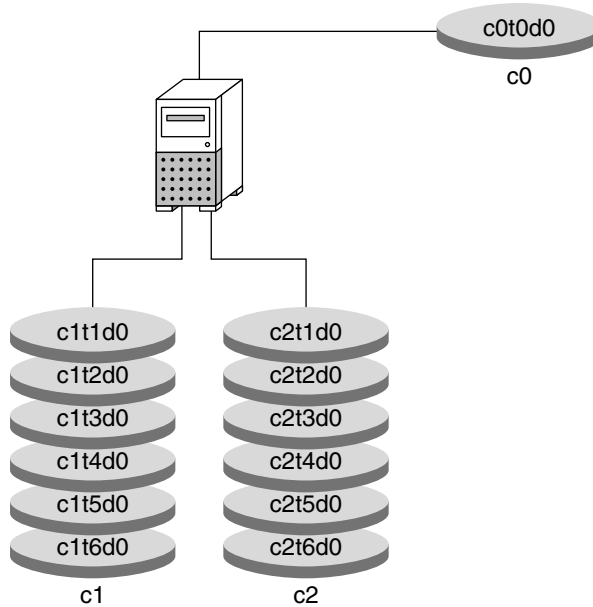
ハードウェア構成

ハードウェアシステムは、次のように構成されています。

- 物理的に分離された 3 つのコントローラがあります (c0 - IDE、c1 - SCSI、および c2 - SCSI)。
- 各 SCSI コントローラは、6 台の 9G バイト内蔵ディスク (c1t1 から c1t6 および c2t1 から c2t6) を含む MultiPack に接続されています。これによってミラー構成が作成されます。
- コントローラ/ターミネータの各ペア (cntn) には、8.49G バイトの使用可能なストレージスペースが存在します。
- ルート (/) ドライブ c0t0d0 上のストレージスペースは、6 つのパーティションに分割されています。

この構成を理解するための代わりの方法を次の図に示します。

図 5-1 ストレージのシナリオの基本的なハードウェア図



初期の物理ストレージ構成

次に、Solaris Volume Manager が構成される前のストレージ構成を示します。

- SCSI コントローラ/ターミネータのペア (*cntn*) には、約 20G バイトのストレージスペースが存在します。
- 各ディスク (たとえば、*c1t1d0*) 上のストレージスペースは、7つのパーティション (*cntnd0s0* から *cntnd0s6*) に分割されています。

ディスクをパーティション分割するには、『[Oracle Solaris の管理: デバイスとファイルシステム](#)』の「ディスクのフォーマット」で説明されている手順に従ってください。

最終的な Solaris Volume Manager 構成

このドキュメントの全体を通して、特定のタスクには特定のシナリオが提供されています。ただし、このドキュメント全体にわたる例をより深く理解できるように、最終的な構成はほぼ、`metastat -p` コマンドで表示される次のような構成になっています。

```
[root@lexicon:~]$ metastat -p
d50 -r c1t4d0s5 c1t5d0s5 c2t4d0s5 c2t5d0s5 c1t1d0s5 c2t1d0s5 -k -i 32b
d1 1 1 c1t2d0s3
d2 1 1 c2t2d0s3
d12 1 1 c1t1d0s0
d13 1 1 c2t1d0s0
d16 1 1 c1t1d0s1
d17 1 1 c2t1d0s1
d25 2 2 c1t1d0s3 c2t1d0s3 -i 32b \
    1 c0t0d0s3
d31 1 2 c1t4d0s4 c2t4d0s4 -i 8192b
d80 -p d70 -o 1 -b 2097152
d81 -p d70 -o 2097154 -b 2097152
d82 -p d70 -o 4194307 -b 2097152
d83 -p d70 -o 6291460 -b 2097152
d84 -p d70 -o 8388613 -b 2097152
d85 -p d70 -o 10485766 -b 2097152
d70 -m d71 d72 1
d71 3 1 c1t3d0s3 \
    1 c1t3d0s4 \
    1 c1t3d0s5
d72 3 1 c2t3d0s3 \
    1 c2t3d0s4 \
    1 c2t3d0s5
d123 -p c1t3d0s6 -o 1 -b 204800
d124 -p c1t3d0s6 -o 204802 -b 204800
d125 -p c1t3d0s6 -o 409603 -b 204800
d126 -p c1t3d0s7 -o 3592 -b 20480
d127 -p c2t3d0s7 -o 3592 -b 1433600
hsp010
hsp014 c1t2d0s1 c2t2d0s1
hsp050 c1t2d0s5 c2t2d0s5
hsp070 c1t2d0s4 c2t2d0s4
```

-p オプションの詳細は、[metastat\(1M\)](#) コマンドを参照してください。

状態データベース (概要)

この章では、状態データベースの複製の概念について説明します。関連タスクの実行については、第7章「状態データベース(タスク)」を参照してください。

この章では、次の内容について説明します。

- 67 ページの「Solaris Volume Manager 状態データベースと複製について」
- 69 ページの「多数決アルゴリズムについて」
- 71 ページの「状態データベースの複製のエラー処理」
- 72 ページの「シナリオ - 状態データベースの複製」

Solaris Volume Manager 状態データベースと複製について

Solaris Volume Manager の状態データベースには、すべてのボリューム、ホットスワップ、ディスクセットの構成とステータスに関する情報が格納されます。Solaris Volume Manager は、冗長性を確保し、システムクラッシュ時のデータベースの破損を防ぐために、複数の状態データベースのコピー (複製) を保持しています (破損するデータベースのコピーはせいぜい1つです)。

状態データベースの複製は、状態データベースのデータが常に有効であることを保証します。状態データベースが更新されると、個々の状態データベースの複製も更新されます。システムがクラッシュした場合に、すべての更新が破損するのを防ぐために、更新は一度に1つずつ行われます。

システムの状態データベースの複製が失われた場合、Solaris Volume Manager はどの状態データベースの複製に有効なデータが格納されているかを判断する必要があります。Solaris Volume Manager は、多数決アルゴリズムを使用して、この情報を判断します。このアルゴリズムでは、過半数 (半数 + 1) の複製が使用可能であり、一致していれば、それらの複製を有効であるとみなします。多数決アルゴリズムの要件のため、ディスク構成を設定するときに、3つ以上の状態データベースの複製を作成する必要があります。3つの状態データベースの複製のうち少なくとも2つが使用可能である限り、合意に達することができます。

ブート時に、Solaris Volume Manager は、破損した状態データベースの複製を無視します。場合によっては、Solaris Volume Manager が破損した状態データベースの複製を再作成しようとしています。そうでなければ、それらは管理者が修正するまで無視されます。基となるスライスにエラーが発生したため、状態データベースの複製が破損した場合は、スライスを修理または交換してから複製を有効にする必要があります。



注意- ファブリック接続されたストレージ、SANなどの、システムに直接接続されていないストレージに、状態データベースの複製を配置しないでください。Solaris Volume Manager をブートできなくなる場合があります。複製は、従来の SCSI または IDE ドライブと同じブートプロセスのポイントで使用できるストレージデバイス上になければなりません。

すべての状態データベースの複製が失われると、理論的には、Solaris Volume Manager のボリュームに格納されているすべてのデータが失われる可能性があります。このため、十分な状態データベースの複製を別々のドライブとコントローラに作成し、深刻な障害を回避することをお勧めします。さらに、最初の Solaris Volume Manager 構成情報とディスクパーティション情報を保存するのも賢明です。

状態データベースの複製をシステムに追加する方法については、第7章「状態データベース(タスク)」を参照してください。状態データベースの複製が失われた場合に回復する方法については、307 ページの「状態データベース複製の障害からの回復」を参照してください。

状態データベースの複製は、RAID-1 ボリュームの再同期領域でも使用されます。ミラーの数に比べて状態データベースの複製の数が少なすぎると、複製の I/O が RAID-1 ボリュームのパフォーマンスに影響を与えることがあります。つまり、ミラーの数が多い場合は、RAID-1 ボリュームあたり少なくとも2つの状態データベースの複製(ディスクセットあたり最大50の複製数)を使用してください。

デフォルトで、ボリューム、ローカルセット、ディスクセットの各状態データベースの複製は、4M バイト(8192 ディスクセクタ)のディスクストレージを占有します。複数所有者ディスクセットの状態データベースの複製のデフォルトのサイズは16M バイトです。

複製は、次のデバイスに格納できます。

- 専用のローカルディスクパーティション
- ボリュームの一部となるローカルパーティション
- UPS ロギングデバイスの一部となるローカルパーティション

複製は、ルート(/)、swap、または/usr スライスには格納できません。複製は、既存のファイルシステムやデータが格納されているスライスにも保存できません。ただし、複製を格納したあとで、同じスライスにボリュームやファイルシステムを置くことができます。

多数決アルゴリズムについて

複製されたデータベースの固有の問題は、どのデータベースが有効で正しいデータを格納しているかを判断するのが難しい場合があることです。この問題を解決するために、Solaris Volume Manager は多数決アルゴリズムを使用します。このアルゴリズムでは、いずれかの複製が有効であると判断されるまでに、大部分の複製の合意が必要です。このアルゴリズムでは、少なくとも最初に3つの複製が作成され、存在している必要があります。3つの複製のうち少なくとも2つが使用可能である限り、合意に達することができます。複製が1つしか存在しておらず、システムがクラッシュした場合、すべてのボリューム構成データが失われる可能性があります。

データを保護するために、Solaris Volume Manager は、すべての状態データベースの複製のうち半数の複製が使用可能でなければ、動作しません。そのため、このアルゴリズムにより、データの破損が防止されます。

多数決アルゴリズムは次のものを提供します。

- システムは、少なくとも状態データベースの複製の半数が使用可能であれば、実行し続けます。
- システムは、使用可能な状態データベースの複製が半数を下回ると、パニックを起こします。
- システムは、過半数(半数+1)の状態データベースの複製が使用可能でなければ、マルチユーザーモードでリブートできません。

十分な状態データベースの複製が使用できない場合は、シングルユーザーモードでブートし、十分な数の破損しているか、失われた複製を削除して、定足数を達成する必要があります。307 ページの「状態データベースの複製数の不足から回復する方法」を参照してください。

注 - 状態データベースの複製の合計数が奇数の場合は、Solaris Volume Manager はその値を2で割り、端数を切り捨てた整数に1を加えることによって過半数を計算します。たとえば、複製が7つあるシステムでは、過半数は4になります(7を2で割ると、3と1/2になり、端数を切り捨てると3になり、1を足すと4になる)。

状態データベースの複製の管理

- 状態データベースの複製のデフォルトのサイズは4Mバイトまたは8192ブロックです。状態データベースの複製は、複製ごとに4Mバイト以上の専用スライス上に作成するようにしてください。ディスクスライスがこれより大きい場合は、状態データベースの複製を格納するために、スライスのサイズを変更する必要があります。スライスサイズの変更については、『Oracle Solaris の管理: デバイスとファイルシステム』の第9章「ディスクの管理(タスク)」を参照してください。

- 単一点障害を避けるため、状態データベースの複製をスライス、ドライブ、コントローラに分散させます。これは、単一のコンポーネントに障害が発生した場合でも、大半の複製を利用可能な状態に保つ必要があるからです。たとえばデバイスの障害のため、複製が失われた場合、Solaris Volume Manager の実行やシステムのリブート時に問題が発生することがあります。Solaris Volume Manager が動作するためには、少なくとも半数の複製が使用可能である必要がありますが、マルチユーザーモードでリブートするためには過半数(半数+1)の複製が使用可能である必要があります。

3つ以上の状態データベースの複製が推奨されます (Solaris Volume Manager ディスクセットあたり最大 50 の複製)。次のガイドラインを推奨します。

- ドライブが1つだけのシステムでは、3つの複製すべてを1つのスライスに置きます。
- ドライブの数が2つから4つのシステムでは、各ドライブに2つずつ複製を置きます。
- ドライブの数が5つ以上のシステムでは、各ドライブに1つずつ複製を置きます。
- 複数のコントローラがある場合、できるだけすべてのコントローラに均等になるように複製を分散してください。この方法によって、コントローラに障害が発生した場合の冗長性が確保できるだけでなく、負荷の分散も可能になります。同じコントローラ上に複数のディスクが存在する場合は、各コントローラで2個以上のディスクに複製を配置します。
- 必要に応じて、RAID-0、RAID-1、またはRAID-5 ボリュームまたはソフトパーティションの一部として使用するスライス上に、状態データベースの複製を作成できます。スライスをボリュームに追加する前に複製を作成する必要があります。Solaris Volume Manager は、スライスの先頭部分を状態データベースの複製用に予約しています。

ボリュームの一部となるスライス上に状態データベースの複製が置かれている場合、ボリュームの容量は、複製によって占有される領域分だけ少なくなります。複製によって使用される領域は、次のシリンダ境界まで切り上げられます。この領域はボリュームによってスキップされます。

- RAID-1 ボリュームは小規模のランダム I/O (データベースで見られるような) に使用します。最高のパフォーマンスを得るため、RAID-1 ボリュームごとに、その RAID-1 ボリュームに接続されていないスライス (および、可能であれば個別のディスクとコントローラ) 上に2つ以上の複製を余分に作成します。
- 状態データベースの複製を、既存のファイルシステムや、ルート (/)、/usr、swap ファイルシステムに作成することはできません。必要であれば、swap からの領域を割り当てて新しいスライスを作成できます (スライス名が使用可能であるとします)。次に、その新しいスライスに状態データベースの複製を配置します。
- 状態データベースの複製は、未使用のスライス上に作成できます。
- いつでも状態データベースの複製をシステムに追加できます。追加の状態データベースの複製によって、Solaris Volume Manager の可用性を確保できます。



注意 - Solstice DiskSuite 製品から Solaris Volume Manager にアップグレードしたときに、スライスが、状態データベースの複製とファイルシステムまたは論理ボリュームの間で共有されている (それぞれが個別のスライス上に置かれていない) 場合は、既存の複製を削除して、同じ場所に新しい複製で置換しないでください。

Solaris Volume Manager の状態データベースの複製のデフォルトのサイズは 8192 ブロックですが、Solstice DiskSuite 製品のデフォルトのサイズは 1034 ブロックです。Solstice DiskSuite 製品で作成されたデフォルトのサイズの状態データベースの複製を削除して、Solaris Volume Manager による新しいデフォルトのサイズの複製を追加する場合は注意してください。共有スライスの残りの部分を占めているファイルシステムの先頭の 7158 ブロックが上書きされ、データが破壊されます。

状態データベースの複製のエラー処理

状態データベースの複製で障害が発生した場合、システムは、少なくとも半数の残りの複製が使用可能であれば動作を続行します。システムは、使用可能な複製が半数を下回ると、パニックを起こします。

システムは、過半数の複製が使用可能であれば、マルチユーザーモードでリポートできます。使用できる複製が過半数に満たない場合は、システムをシングルユーザーモードでリポートし、`metadb` コマンドを使って使用不能な複製を削除する必要があります。

たとえば、4つの複製を使用しているとします。システムは、2つの複製(半数)が使用可能であれば動作を続けます。ただし、システムをリポートするには、3つの複製(半数+1)が使用可能でなければなりません。

2ディスク構成では、各ディスクに必ず2つ以上の複製を作成します。たとえば、2ディスク構成で複製を3つしか作成しないとします(最初のディスクに2つの複製、2つ目のディスクに1つの複製)。2つの複製のあるディスクに障害が発生すると、残りのディスクの複製が1つだけになるため、システムはパニックを起こします。これは、複製の半数に足りません。

注-2ディスク構成で各ディスクに2つずつ複製を作成すれば、一方のディスクに障害が発生しても、Solaris Volume Manager は機能し続けます。しかし、システムのリポートには過半数の複製が必要なため、リポートできません。

状態データベースの複製を格納するスライスで障害が発生した場合、構成の残りの部分は動作し続けるはずですが、Solaris Volume Manager は、ブート時に、過半数の有効な状態データベースの複製が使用可能であれば、有効な状態データベースを探します。

状態データベースの複製を手動で修復するか、有効にすると、Solaris Volume Manager は有効なデータを使ってそれらを更新します。

シナリオ—状態データベースの複製

状態データベースの複製は、Solaris Volume Manager 構成全体に関する冗長データを提供します。次の例は、第5章「Solaris Volume Manager の構成と使用 (シナリオ)」に示すシナリオのサンプルシステムに基づいています。この例では、適切な冗長性を得るために、状態データベースの複製を分散する方法を示しています。

サンプルシステムには1つの内蔵 IDE コントローラとドライブ、さらに2台の SCSI コントローラがあります。SCSI コントローラにはそれぞれ6つのディスクが接続されています。3つのコントローラによって、単一点障害を防止するように、システムを構成できます。2つのコントローラしかないシステムでは、Solaris Volume Manager に関する単一点障害を防止することはできません。複製を3つのすべてのコントローラと各コントローラの少なくとも1つ(可能であれば2つ)のディスクに均一に分散することによって、システムはどのようなハードウェアの単一点障害にも耐えることができます。

最小限の構成では、1つの状態データベースの複製をルートディスクのスライス7に置き、追加の複製を他の2つの各コントローラの1つのディスクのスライス7に置くことができます。きわめて可能性の低いメディアの障害に対して保護するために、ルートディスクにもう1つの複製を追加すると、各コントローラの2つのディスク上の2つの複製(合計で6つの複製)によって十分な安全性が確保されます。

さらに完全性を得るには、2つのミラーのそれぞれの側の6つのディスクに均等に分散された12個の複製を追加します。この構成の結果、複製はルートディスクに2つ、各 SCSI コントローラ上のディスクに分散された8つずつで合計18になります。

状態データベース (タスク)

この章では、Solaris Volume Manager の状態データベースの複製に関連するタスクの実行について説明します。これらのタスクに関連する概念については、第6章「状態データベース (概要)」を参照してください。

状態データベースの複製 (タスクマップ)

次のタスクマップは、Solaris Volume Manager の状態データベースの複製を管理するために必要な手順を示します。

タスク	説明	参照先
状態データベースの複製を作成する	Solaris Volume Manager の GUI または <code>metadb -a</code> コマンドを使用して、状態データベースの複製を作成します。	74 ページの「状態データベースの複製を作成する方法」
状態データベースの複製のステータスをチェックする	Solaris Volume Manager の GUI または <code>metadb</code> コマンドを使用して、既存の複製のステータスをチェックします。	76 ページの「状態データベースの複製のステータスをチェックする方法」
状態データベースの複製を削除する	Solaris Volume Manager の GUI または <code>metadb -d</code> コマンドを使用して、状態データベースの複製を削除します。	77 ページの「状態データベースの複製を削除する方法」

状態データベースの複製の作成



注意 - Solstice DiskSuite 製品から Solaris Volume Manager にアップグレードしている、状態データベースの複製とファイルシステムまたは論理ボリュームの間でスライスが共有されている (それぞれが異なるスライス上に置かれていない) 場合は、既存の複製を削除して同じ場所に新しいデフォルトの複製を作成してはなりません。

Solaris Volume Manager の状態データベースの複製のデフォルトのサイズは 8192 ブロックですが、Solstice DiskSuite 製品のデフォルトのサイズは 1034 ブロックです。Solstice DiskSuite 製品で作成されたデフォルトのサイズの状態データベースの複製を削除して、Solaris Volume Manager による新しいデフォルトのサイズの複製を追加する場合は注意してください。共有スライスの残りの部分を占めているファイルシステムの先頭の 7158 ブロックが上書きされ、データが破壊されます。



注意 - ファブリック接続されたストレージ、SAN などの、システムに直接接続されていないストレージに、状態データベースの複製を配置しないでください。Solaris Volume Manager をブートできなくなる場合があります。複製は、従来の SCSI または IDE ドライブと同じブートプロセスのポイントで使用できるストレージデバイス上になければなりません。

▼ 状態データベースの複製を作成する方法

始める前に [51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」](#)を確認してください。

- 1 スーパーユーザーになります。
- 2 状態データベースの複製を作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「状態データベース複製」ノードを開きます。「アクション」、「複製を作成」の順に選択し、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metadb` コマンドを使用します。[metadb\(1M\)](#) を参照してください。

```
# metadb -a -c number -l length-of-replica -f ctds-of-slice
```

-a 状態データベースの複製を追加または作成することを指定します。

-f 複製が存在しなくても、強制的に操作を実行することを指定します。最初の複製を強制的に作成するには、-f を使用します。

-c number 特定のスライスに追加する複製の数を指定します。

<code>-l length-of-replica</code>	新しい複製のサイズをブロック数で指定します。デフォルトのサイズは8192です。このサイズは、数千の論理ボリュームを持つ構成を含め、事実上すべての構成に適しています。
<code>ctds-of-slice</code>	複製を格納するコンポーネントの名前を指定します。

注 - コマンド行にオプションを指定しないで `metadb` コマンドを入力すると、すべての状態データベース複製のステータスが報告されます。

例 7-1 最初の状態データベース複製を作成する

```
# metadb -a -f c0t0d0s7
# metadb
      flags      first blk      block count
...
  a      u          16          8192          /dev/dsk/c0t0d0s7
```

最初の状態データベース複製を作成するには、`-f` オプションを `-a` オプションとともに使用する必要があります。`-a` オプションは、状態データベースの複製をシステムに追加します。`-f` オプションは、最初の複製を強制的に作成します (システムに補助的な複製を追加する場合は省略可能)。

例 7-2 2つの状態データベース複製を同じスライスに追加する

```
# metadb -a -c 2 c1t3d0s1
# metadb
      flags      first blk      block count
...
  a      u          16          8192          /dev/dsk/c1t3d0s1
  a      u          8208         8192          /dev/dsk/c1t3d0s1
```

`-a` オプションは、状態データベースの複製をシステムに追加します。`-c 2` オプションは、指定したスライスに2つの複製を格納します。`metadb` コマンドは、複製がアクティブであるかどうかを調べ、`metadb` コマンド出力の `a` フラグで示します。

例 7-3 特定のサイズの状態データベース複製を追加する

既存の状態データベースの複製を置き換える場合は、複製のサイズを指定しなければならない場合があります。特に、ファイルシステムとスライスを共有している状態データベースの複製がすでに存在している場合は (たとえば、Solstice DiskSuite 製品からアップグレードした場合など)、既存の複製を同じサイズの複製で置き換えるか、別の場所に新しい複製を作成する必要があります。

```
# metadb -a -c 3 -l 1034 c0t0d0s7
# metadb
```

```

                flags          first blk      block count
...
a      u      16              1034          /dev/dsk/c0t0d0s7
a      u      1050           1034          /dev/dsk/c0t0d0s7
a      u      2084           1034          /dev/dsk/c0t0d0s7

```

-a オプションは、状態データベースの複製をシステムに追加します。-l オプションは、追加する複製の長さをブロック数で指定します。

状態データベースの複製の保守

▼ 状態データベースの複製のステータスをチェックする方法

- 1 スーパーユーザーになります。
- 2 状態データベースの複製のステータスをチェックするには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「状態データベースの複製」ノードを開いて、存在するすべての状態データベース複製を表示します。詳細は、オンラインヘルプを参照してください。
 - `metadb` コマンドを使用して、状態データベースの複製のステータスを表示します。-i オプションを指定すると、ステータスフラグの説明が表示されます(次の例を参照)。[metadb\(1M\)](#) を参照してください。

例 7-4 すべての状態データベース複製のステータスをチェックする

```

# metadb -i
      flags          first blk      block count
a m p luo          16              8192          /dev/dsk/c0t0d0s7
a p luo            8208           8192          /dev/dsk/c0t0d0s7
a p luo           16400          8192          /dev/dsk/c0t0d0s7
a p luo            16              8192          /dev/dsk/c1t3d0s1
W p l              16              8192          /dev/dsk/c2t3d0s1
a p luo            16              8192          /dev/dsk/c1t1d0s3
a p luo            8208           8192          /dev/dsk/c1t1d0s3
a p luo           16400          8192          /dev/dsk/c1t1d0s3
r - replica does not have device relocation information
o - replica active prior to last mddb configuration change
u - replica is up to date
l - locator for this replica was read successfully
c - replica's location was in /etc/lvm/mddb.cf
p - replica's location was patched in kernel
m - replica is master, this is replica selected as input

```

```

W - replica has device write errors
a - replica is active, commits are occurring to this replica
M - replica had problem with master blocks
D - replica had problem with data blocks
F - replica had format problems
S - replica is too small to hold current data base
R - replica had device read errors

```

ステータスの後にすべてのフラグの説明が表示されます。デバイス名の前の文字はデバイスのステータスを表します。大文字は障害ステータスを示します。小文字は「正常」ステータスを示します。

▼ 状態データベースの複製を削除する方法

Solaris Volume Manager 構成を保守するために、状態データベースの複製を削除しなければならない場合があります。たとえば、ディスクドライブを交換する場合、ドライブを取り外す前に、状態データベースの複製を削除します。そうしないと、Solaris Volume Manager からエラーが報告されます。

- 1 スーパーユーザーになります。
- 2 状態データベースの複製を削除するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「状態データベースの複製」ノードを開いて、存在するすべての状態データベース複製を表示します。削除する複製を選択してから、「編集」、「削除」の順に選択して複製を削除します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metadb` コマンドを使用します。

```
# metadb -d -f ctds-of-slice
```

`-d` 状態データベースの複製を削除することを指定します。

`-f` 複製が存在しなくても、強制的に操作を実行することを指定します。

`ctds-of-slice` 複製が格納されているコンポーネントの名前を指定します。

削除したい状態データベースの複製が格納されている各スライスを指定する必要があります。詳細は、`metadb(1M)` のマニュアルページを参照してください。

例 7-5 状態データベースの複製を削除する

```
# metadb -d -f c0t0d0s7
```

この例では、スライスから最後の複製を削除します。

システム上にある最後の複製を強制的に削除するには、`-f` オプションを指定する必要があります。

RAID-0(ストライプと連結)ボリューム(概要)

この章では、Solaris Volume Manager で使用できる RAID-0(ストライプと連結) ボリュームについて説明します。関連するタスクの詳細については、第9章「RAID-0(ストライプおよび連結)ボリューム(タスク)」を参照してください。

この章では、次の内容について説明します。

- 79 ページの「RAID-0 ボリュームの概要」
- 87 ページの「RAID-0 ボリュームを作成するための背景情報」
- 88 ページの「シナリオ-RAID-0 ボリューム」

RAID-0 ボリュームの概要

RAID-0 ボリュームは、スライスまたはソフトパーティションから構成されています。これらのボリュームでは、ディスクのストレージ容量を拡張できます。それらは直接使用することも、または RAID-1(ミラー) ボリュームやソフトパーティションの構成要素としても使用できます。RAID-0 ボリュームは3種類あります。

- ストライプボリューム
- 連結ボリューム
- 連結ストライプボリューム

注- 「コンポーネント」とは、別の論理ボリュームで使われているスライスからソフトパーティションまで、すべてのデバイスを表します。

ストライプボリュームは、ボリューム内のすべてのコンポーネントに均一にデータを分散し、連結ボリュームは、最初に使用可能なコンポーネントに、それがいっぱいになるまでデータを書き込んでから、次の使用可能なコンポーネントに移動します。連結ストライプボリュームは単に、コンポーネントを追加して、その元の構成から拡張されたストライプボリュームです。

RAID-0 ボリュームでは、ディスクのストレージ容量を迅速かつ簡単に拡張できます。欠点は、これらのボリュームは RAID-1 や RAID-5 ボリュームと異なり、データ冗長性を提供しないことです。RAID-0 ボリュームの1つのコンポーネントで障害が発生すると、データが失われます。

ストライプボリュームへの順次 I/O 操作の場合、Solaris Volume Manager は、最初のコンポーネントのすべてのブロックをブロックのセグメント(「インターレース」と呼ばれる)で読み取り、次に2番目のコンポーネントのすべてのブロックをブロックのセグメントで読み取るというように繰り返します。

連結ボリュームへの順次 I/O 操作の場合、Solaris Volume Manager は、最初のコンポーネントのすべてのブロックを読み取り、次に2番目のコンポーネントのすべてのブロックを読み取るというように繰り返します。

連結ボリュームでもストライプボリュームでも、すべての I/O 操作が並列で行なわれます。

1つのスライスを含む RAID-0 ボリュームは、任意のファイルシステムに使用できます。

複数のコンポーネントを含む RAID-0 ボリュームは、次を除く任意のファイルシステムに使用できます。

- ルート (/)
- /usr
- swap
- /var
- /opt
- オペレーティングシステムのアップグレードまたはインストール時にアクセスされる任意のファイルシステム

注-ルート (/)、/usr、swap、/var、または /opt をミラー化する場合、サブミラーとして機能する1方向の連結またはストライプ(単一スライスの連結)にファイルシステムを置きます。この1方向の連結を、同様に連結である別のサブミラーによってミラー化します。

RAID-0(ストライプ)ボリューム

RAID-0(ストライプ)ボリュームは、データを1つまたは複数のコンポーネントに配置するボリュームです。ストライプでは、同じサイズのデータのセグメントが2つ以上のコンポーネントに交互に配置され、1つの論理ストレージユニットが形成されます。これらのセグメントは、ラウンドロビン方式でインターリーブされるため、各コンポーネントから交互に組み合わせられた領域が作成され、事実上、トランプのようにシャッフルされます。

注-ストライプボリュームの容量を増やすには、連結ストライプボリュームを構築する必要があります。84 ページの「RAID-0(連結ストライプ)ボリューム」を参照してください。

ストライプにより、複数のコントローラが同時にデータにアクセスできますが、これは「並列アクセス」とも呼ばれます。並列アクセスでは、ボリューム内のすべてのディスクでほとんどの時間が I/O リクエストの処理でビジーになるため、I/O スループットを向上させることができます。

既存のファイルシステムをストライプに直接変換することはできません。ストライプボリュームに既存のファイルシステムを配置するには、ファイルシステムをバックアップし、ボリュームを作成してから、ストライプボリュームにファイルシステムを復元する必要があります。

RAID-0(ストライプ)ボリュームのインターレース値

インターレースは、ストライプボリューム上の論理データセグメントの K バイト、M バイト、またはブロック単位のサイズです。アプリケーションによって、構成のパフォーマンスを向上させるインターレース値が異なることがあります。パフォーマンスの向上は、I/O リクエストを管理する複数のディスクアームを使用することによって達成されます。I/O リクエストがインターレースのサイズより大きい場合、パフォーマンスが向上する可能性があります。

注-RAID-5 ボリュームでもインターレース値が使われます。詳細については、167 ページの「RAID-5 ボリュームの概要」を参照してください。

ストライプボリュームを作成する場合、インターレース値を設定するか、Solaris Volume Manager の 16K バイトのデフォルトのインターレース値を使用します。ストライプボリュームを作成したら、インターレース値は変更できません。ただし、ボリューム上のデータをバックアップし、ストライプボリュームを削除して、新しいインターレース値で新しいストライプボリュームを作成してから、データを復元することができます。

シナリオ-RAID-0(ストライプ)ボリューム

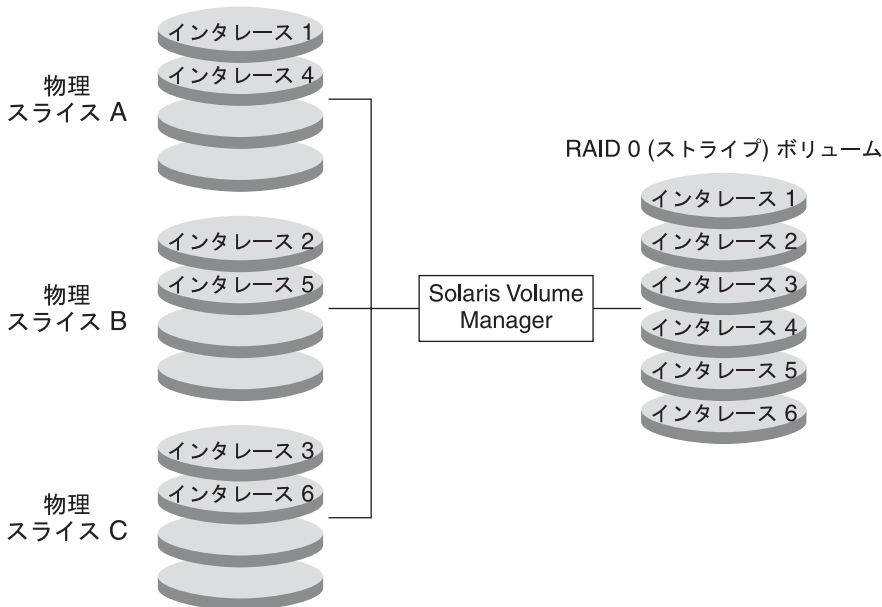
図 8-1 に、3つのコンポーネント(スライス)から構築されたストライプボリュームを示します。さらに、インターレースサイズに従い、ラウンドロビン手法を使用して、データがボリュームコンポーネントに書き込まれるしくみも示しています。

Solaris Volume Manager がストライプボリュームのコンポーネントにデータを書き込む場合、インターレース幅のデータブロックをディスク A(インターレース 1)、ディスク B(インターレース 2)、およびディスク C(インターレース 3)に書き込み

ます。さらに、Solaris Volume Managerはこのパターンを繰り返して、ディスク A(インターレース4)、ディスク B(インターレース5)、ディスク C(インターレース6)というように書き込みます。

インターレース値は、毎回データがスライスに書き込まれるサイズを設定します。ストライプボリュームの総容量は、コンポーネントの数に最小のコンポーネントのサイズを掛けた値に等しくなります。(次の例の各スライスが2Gバイトの場合、ボリュームは6Gバイトになります。)

図 8-1 RAID-0(ストライプ)ボリュームの例



RAID-0(連結)ボリューム

RAID-0(連結)ボリュームは、データがコンポーネント全体に連続して隣接して編成され、1つの論理ストレージユニットを形成しているボリュームです。

連結ボリュームを使用して、複数のコンポーネントの容量を組み合わせることで、ストレージ容量を増やします。ストレージの要求の拡大に応じて、連結ボリュームにコンポーネントを追加できます。

連結ボリュームでは、ストレージ容量とファイルシステムのサイズをオンラインで動的に拡張できます。連結ボリュームでは、他のコンポーネントが現在アクティブであっても、コンポーネントを追加できます。

また、連結ボリュームでは、システムを停止する必要なく、任意のアクティブなマウントされている UFS ファイルシステムを拡張することもできます。一般に、連結ボリュームの総容量は、ボリューム内のすべてのコンポーネントの合計サイズに等しくなります。連結ボリュームに状態データベースの複製を格納するスライスが含まれている場合、ボリュームの総容量は、コンポーネントの合計から、その複製用に予約されている領域を引いた値になります。

単一のコンポーネントから連結ボリュームを作成することもできます。後で、ストレージを増やす必要があるときに、ボリュームにコンポーネントを追加できます。

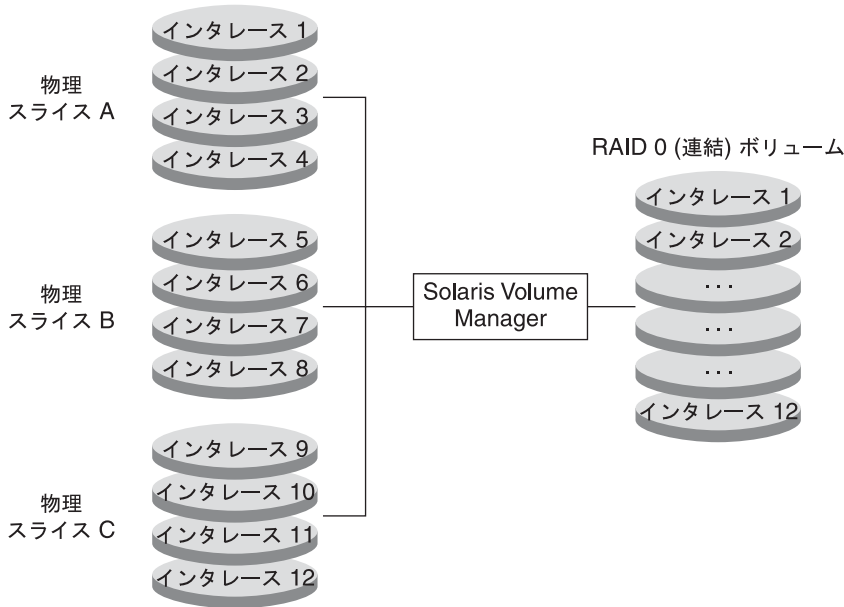
注-ルート (/)、swap、/usr、/opt、または/var のファイルシステムをミラー化する場合は、連結ボリュームを使用してこれらをカプセル化する必要があります。

シナリオ - RAID-0 (連結) ボリューム

図 8-2 に、3つのコンポーネント(スライス)から構築された連結ボリュームを示します。さらに、インターレースサイズに従って、各スライスに順番に、データがボリュームコンポーネントに書き込まれるしくみも示しています。

データブロックは、スライス A からコンポーネント全体に順番に書き込まれます。スライス A は、論理データブロック 1 から 4 を格納するものと想定できません。ディスク B は、論理データブロック 5 から 8 を格納します。ディスク C は、論理データブロック 9 から 12 を格納します。ボリュームの総容量は、3つのスライスを組み合わせた容量になります。各スライスが 2G バイトの場合、ボリューム全体の容量は 6G バイトになります。

図 8-2 RAID-0(連結)ボリュームの例



RAID-0(連結ストライプ)ボリューム

RAID-0(連結ストライプ)ボリュームは、コンポーネント(ストライプ)を追加して拡張されたストライプです。

ストライプレベルで、連結ストライプボリュームのインターレース値を設定するには、Solaris 管理コンソール内の拡張ストレージツールまたは `metattach -i` コマンドを使用します。連結ストライプボリューム内の各ストライプに、固有のインターレース値を指定できます。最初から連結ストライプボリュームを作成する場合、特定のストライプのインターレース値を指定しないと、前にボリュームに追加されたストライプからインターレース値が継承されます。

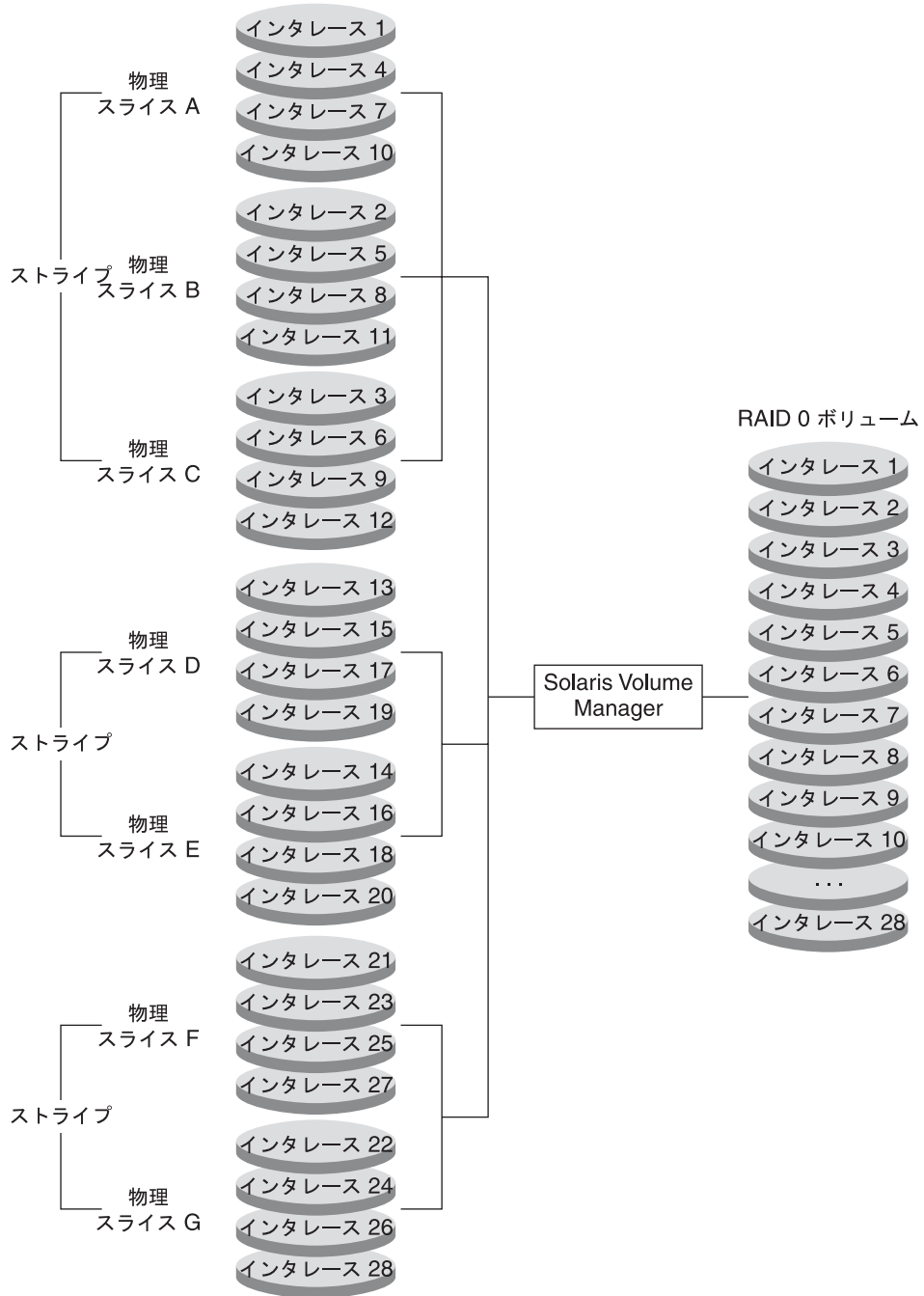
例 — RAID-0(連結ストライプ)ボリューム

図 8-3 に、3つのストライプの連結である連結ストライプボリュームを示します。

最初のストライプは、16K バイトのインターレース値で、スライス A からスライス C の3つのスライスから構成されています。2番目のストライプは、スライス D とスライス E の2つのスライスから構成され、32K バイトのインターレース値を使用します。3番目のストライプは、スライス F とスライス G の2つのスライスから構成されています。3番目のストライプにはインターレース値が指定されていないため、以前に追加されたストライプから値が継承され、この例では32K バイトになります。最初のストライプに空きがなくなるまで、順次データブロックが追加されます。次

に、データブロックが2番目のストライプに追加されます。このストライプにも空きがなくなったら、3番目のストライプにデータブロックが追加されます。各ストライプ内で、データブロックは、指定したインターレース値に従ってインターリーブされます。

図 8-3 RAID-0(連結ストライプ)ボリュームの例



RAID-0 ボリュームを作成するための背景情報

RAID-0 ボリュームの要件

RAID-0 ボリュームを操作する場合は、次のことを考慮します。

- それぞれ異なるコントローラ上にあるコンポーネントを使用すると、実行可能な同時読み取りと書き込みの数が増えます。
- 既存のファイルシステムやデータからストライプを作成しないでください。そうすると、データが破壊されます。代わりに、連結を使用します。(既存のデータからストライプを作成できますが、データをダンプし、ボリュームに復元する必要があります。)
- ストライプには同じサイズのディスクコンポーネントを使用します。異なるサイズのコンポーネントをストライプ化すると、ディスク領域が無駄になります。
- システムやアプリケーションによって行なわれる I/O リクエストに合わせて、ストライプのインターレース値を設定します。
- ストライプや連結には、複製されたデータが格納されていないため、それらのボリュームのコンポーネントで障害が発生した場合は、コンポーネントを交換し、ストライプまたは連結を再作成し、バックアップからデータを復元する必要があります。
- ストライプまたは連結を再作成する場合は、障害のあるコンポーネントと少なくとも同じサイズの交換用のコンポーネントを使用します。

RAID-0 ボリュームのガイドライン

- 連結は、ストライプより使用する CPU サイクルが少なく、小さなランダム I/O や均一の I/O 分散に適しています。
- 可能な限り、ストライプや連結のコンポーネントは、別々のコントローラやバスに分散します。それぞれ異なるコントローラ上にあるストライプを使用すると、実行可能な同時読み取りと書き込みの数が増えます。
- ストライプが障害のあるコントローラ上に定義されており、システムで別のコントローラが使用できる場合、ディスクをそのコントローラに移動し、ストライプを再定義することによって、ストライプを新しいコントローラに「移動」できます。
- ストライプの数: ストライプ化について考える別の方法は、まずパフォーマンス要件を判断することです。たとえば、選択したアプリケーションに 10.4M バイト/秒のパフォーマンスが必要で、各ディスクのパフォーマンスがおおよそ 4M バイト/秒であるとし、次の公式に基づいて、ストライプ化するために必要なディスクスピンドル数を判断します。

10.4M バイト/秒 / 4M バイト/秒 = 2.6

従って、並列でI/O操作を実行できる3つのディスクが必要です。

シナリオ-RAID-0ボリューム

RAID-0ボリュームは、複雑なストレージ構成を作成したり、ミラーを構築したりするための基本構成要素を提供します。次の例では、[第5章「Solaris Volume Managerの構成と使用\(シナリオ\)」](#)に説明するシナリオを利用して、RAID-0ボリュームで大きなストレージ領域を提供し、ルート(/)を含む既存のファイルシステムのミラーを構築する方法を説明します。

シナリオのサンプルシステムには、比較的小さな(9Gバイト)ディスクのコレクションがありますが、特定のアプリケーションでもっと大きなストレージ領域を必要とする可能性があります。大きな領域を作成する(およびパフォーマンスを向上させる)ために、複数のディスクにまたがるストライプを作成できます。たとえば、ディスク `c1t1d0`、`c1t2d0`、`c1t3d0`、`c2t1d0`、`c2t2d0`、および `c2t3d0` のそれぞれを、ディスク全体にまたがるスライス 0 でフォーマットできます。これにより、1つのコントローラからの3つすべてのディスクを含むストライプは、約27Gバイトのストレージを提供し、アクセスを高速化できます。2番目のコントローラからの2番目のストライプは、[第11章「RAID-1\(ミラー\)ボリューム\(タスク\)」](#) および特に [110ページ](#)の「[シナリオ-RAID-1ボリューム\(ミラー\)](#)」に説明するように、冗長として使用できます。

RAID-0(ストライプおよび連結) ボリューム(タスク)

この章では、RAID-0 ボリュームに関連したタスクについて説明します。関連する概念については、第8章「RAID-0(ストライプと連結) ボリューム(概要)」を参照してください。

RAID-0 ボリューム(タスクマップ)

次のタスクマップに、Solaris Volume Manager の RAID-0 ボリュームを管理するために必要な手順を示します。

タスク	説明	参照先
RAID-0(ストライプ) ボリュームを作成する	新しいボリュームを作成するには、Solaris Volume Manager GUI または <code>metainit</code> コマンドを使用します。	90 ページの「RAID-0(ストライプ) ボリュームを作成する方法」
RAID-0(連結) ボリュームを作成する	新しいボリュームを作成するには、Solaris Volume Manager GUI または <code>metainit</code> コマンドを使用します。	92 ページの「RAID-0(連結) ボリュームを作成する方法」
ストレージスペースを拡張する	既存のファイルシステムを拡張するには、Solaris Volume Manager GUI または <code>metainit</code> コマンドを使用します。	93 ページの「既存のデータのストレージ容量を拡張する方法」
既存の RAID-0 ボリュームを拡張する	既存のボリュームを拡張するには、Solaris Volume Manager GUI または <code>metattach</code> コマンドを使用します。	95 ページの「既存の RAID-0 ボリュームを拡張する方法」
RAID-0 ボリュームを削除する	ボリュームを削除するには、Solaris Volume Manager GUI または <code>metaclear</code> コマンドを使用します。	97 ページの「RAID-0 ボリュームを削除する方法」

RAID-0 (ストライプ) ボリュームの作成



注意-既存のファイルシステムやデータからストライプを作成しないでください。それを行うと、データが破棄されます。既存のデータからストライプを作成するには、そのデータをバックアップし、ストライプボリュームを作成してから、データをそのボリュームに戻す必要があります。



注意-32ビットカーネルで Solaris ソフトウェアを実行することを予定している場合は、1Tバイトを超えるボリュームを作成しないでください。さらに、Solaris 9 4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、1Tバイトを超えるボリュームを作成しないでください。Solaris Volume Manager での大容量ボリュームのサポートの詳細は、52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」を参照してください。

▼ RAID-0 (ストライプ) ボリュームを作成する方法

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 87 ページの「RAID-0 ボリュームを作成するための背景情報」を確認してください。

- ストライプボリュームを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「アクション」、「ボリュームを作成」の順に選択したあと、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes
  components-per-stripe
  component-names
  [ -i interlace]
```

volume-name 作成するボリュームの名前を指定します。ボリュームの命名については、47 ページの「ボリューム名」を参照してください。

number-of-stripes 作成するストライプの数を指定します。

components-per-stripe 各ストライプに含めるコンポーネントの数を指定します。

<i>component-names</i>	使用されるコンポーネントの名前を指定します。複数のコンポーネントが使用される場合は、各コンポーネントをスペースで区切ります。
<i>-i interlace</i>	ストライプに使用するインタレース幅を指定します。このインタレース幅は、値のあとにKバイトを示す「k」、Mバイトを示す「m」、ブロックを示す「b」のいずれかを付けて指定します。指定されたインタレースが16ブロック未満であったり、100Mバイトを超えていたりしてはいけません。デフォルトのインタレース幅は16Kバイトです。

詳細は、次の例および [metainit\(1m\)](#) のマニュアルページを参照してください。

例 9-1 3つのスライスから成る RAID-0(ストライプ)ボリュームの作成

```
# metainit d20 1 3 c0t1d0s2 c0t2d0s2 c0t3d0s2
d20: Concat/Stripe is setup
```

この例は、ストライプ `d20` が1つのストライプ(数字 1)で構成されることを示しています。このストライプは、3つのスライス(数字 3)で構成されます。インタレース値が指定されていないため、このストライプは16Kバイトのデフォルト値を使用します。システムは、ボリュームが設定されていることを確認します。

例 9-2 インタレース値が32Kバイトの2つのスライスから成る RAID-0(ストライプ)ボリュームの作成

```
# metainit d10 1 2 c0t1d0s2 c0t2d0s2 -i 32k
d10: Concat/Stripe is setup
```

この例は、ストライプ `d10` が1つのストライプ(数字 1)で構成されることを示しています。このストライプは、2つのスライス(数字 2)で構成されます。`-i` オプションは、インタレース値を32Kバイトに設定します。(インタレース値が8Kバイト未満であったり、100Mバイトを超えていたりしてはいけません。)システムは、ボリュームが設定されていることを確認します。

参照 新しく作成されたストライプをファイルシステム用に準備するには、『[Oracle Solarisの管理: デバイスとファイルシステム](#)』の第15章「[ファイルシステムの作成およびマウント\(タスク\)](#)」を参照してください。データベースなどの一部のアプリケーションは、ファイルシステムを使用しません。これらのアプリケーションは、代わりにraw デバイスを使用します。このアプリケーションは、raw デバイスにアクセスするための独自の方法を備えている必要があります。

RAID-0 (連結) ボリュームの作成

▼ RAID-0 (連結) ボリュームを作成する方法



注意 - 32ビットカーネルで Solaris ソフトウェアを実行することを予定している場合は、1Tバイトを超えるボリュームを作成しないでください。さらに、Solaris 9 4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、1Tバイトを超えるボリュームを作成しないでください。Solaris Volume Manager でのマルチテラバイトボリュームの詳細は、52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」を参照してください。

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 87 ページの「RAID-0 ボリュームを作成するための背景情報」を確認してください。

- 連結ボリュームを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「アクション」、「ボリュームを作成」の順に選択したあと、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes
components-per-stripe
component-names
```

<code>volume-name</code>	作成するボリュームの名前を指定します。
<code>number-of-stripes</code>	作成するストライプの数を指定します。
<code>components-per-concatenation</code>	各連結に含めるコンポーネントの数を指定します。
<code>component-names</code>	使用されるコンポーネントの名前を指定します。複数のコンポーネントが使用される場合は、各コンポーネントをスペースで区切ります。

詳細は、次の例および `metainit(lm)` のマニュアルページを参照してください。

例 9-3 1つのスライスから成る連結の作成

```
# metainit d25 1 1 c0t1d0s2
d25: Concat/Stripe is setup
```

この例は、連結 d25 の作成を示しています。この連結は1つのストライプ (最初の数字 1) で構成され、そのストライプが1つのスライス (スライスの前の 2 番目の数字 1) で構成されます。システムは、ボリュームが設定されていることを確認します。

この例は、既存のデータを安全にカプセル化できる連結を示しています。

例 9-4 4つのスライスから成る連結の作成

```
# metainit d40 4 1 c0t1d0s2 1 c0t2d0s2 1 c0t2d0s3 1 c0t2d1s3
d40: Concat/Stripe is setup
```

この例は、連結 d40 の作成を示しています。この連結は4つのストライプ (数字 4) で構成され、各ストライプが1つのスライス (各スライスの前の数字 1) で構成されます。システムは、ボリュームが設定されていることを確認します。

参照 新しく作成された連結をファイルシステム用に準備するには、『Oracle Solaris の管理: デバイスとファイルシステム』の第 15 章「ファイルシステムの作成およびマウント (タスク)」を参照してください。

ストレージ容量の拡張

ファイルシステムにストレージ容量を追加するには、連結ボリュームを作成します。既存のストライプにストレージ容量を追加するには、連結ストライプボリュームを作成します。

▼ 既存のデータのストレージ容量を拡張する方法



注意 - 32 ビットカーネルで Solaris ソフトウェアを実行することを予定している場合は、1T バイトを超えるボリュームを作成しないでください。さらに、Solaris 9.4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、1T バイトを超えるボリュームを作成しないでください。Solaris Volume Manager でのマルチテラバイトボリュームのサポートの詳細は、52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」を参照してください。

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 87 ページの「RAID-0 ボリュームを作成するための背景情報」を確認してください。

- 1 ファイルシステムをマウント解除します。

```
# umount /filesystem
```

- 2 連結を作成するには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「アクション」、「ボリュームを作成」の順に選択したあと、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes
    components-per-stripe
    component-names
```

`volume-name` 作成するボリュームの名前を指定します。

`number-of-stripes` 作成するストライプの数を指定します。

`components-per-stripe` 各ストライプに含めるコンポーネントの数を指定します。

`component-names` 使用されるコンポーネントの名前を指定します。複数のコンポーネントが使用される場合は、各コンポーネントをスペースで区切ります。

詳細は、[metainit\(1m\)](#) のマニュアルページを参照してください。

- 3 `/etc/vfstab` ファイルを編集して、ファイルシステムが連結の名前を参照するようにします。

- 4 ファイルシステムを再マウントします。

```
# mount /filesystem
```

例 9-5 連結を作成することによるファイルシステムの拡張

```
# umount /docs
# metainit d25 2 1 c0t1d0s2 1 c0t2d0s2
d25: Concat/Stripe is setup
    (Edit the /etc/vfstab file so that the file system references the volume d25 instead of slice c0t1d0s2)
# mount /docs
```

この例は、2つのスライス `/dev/dsk/c0t1d0s2` (`/docs` 上にマウントされたファイルシステムを含む) と `/dev/dsk/c0t2d0s2` からの連結 `d25` の作成を示しています。最初にファイルシステムをアンマウントする必要があります。 `metainit` コマンドの最初のスライスは、ファイルシステムを含むスライスである必要があることに注意してください。そうでない場合は、データが破損します。

次に、`/etc/vfstab` ファイル内のファイルシステムのエントリが、その連結を参照するように変更(はじめての場合は入力)されます。たとえば、最初、`/etc/vfstab` ファイルに次の行があったとします。

```
/dev/dsk/c0t1d0s2 /dev/rdisk/c0t1d0s2 /docs ufs 2 yes -
```

この行は、次のように変更されるはずですが。

```
/dev/md/dsk/d25 /dev/md/rdisk/d25 /docs ufs 2 yes -
```

最後に、ファイルシステムが再マウントされます。

参照 UFS ファイルシステムの場合は、連結に対して `growfs` コマンドを実行します。247 ページの「[ファイルシステムを拡張する方法](#)」を参照してください。

データベースなどの一部のアプリケーションは、ファイルシステムを使用しません。データベースなどのアプリケーションは raw 連結を使用するため、この連結を認識するか、または追加された容量を拡張するための独自の方法を備えている必要があります。

▼ 既存の RAID-0 ボリュームを拡張する方法

連結ストライプを使用すると、既存のストライプを拡張できます。たとえば、ストライプのストレージ容量が不足している場合は、そのストライプを連結ストライプに変換できます。これにより、データのバックアップや復元を行わなくてもストレージ容量を拡張できるようになります。

この手順では、既存のストライプにストライプを追加していることを前提にしています。



注意 - 32 ビットカーネルで Solaris ソフトウェアを実行することを予定している場合は、1T バイトを超えるボリュームを作成しないでください。さらに、Solaris 9 4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、1T バイトを超えるボリュームを作成しないでください。Solaris Volume Manager でのマルチテラバイトのサポートの詳細は、52 ページの「[Solaris Volume Manager でのマルチテラバイトのサポートの概要](#)」を参照してください。

始める前に 51 ページの「[Solaris Volume Manager コンポーネントを作成するための前提条件](#)」および 87 ページの「[RAID-0 ボリュームを作成するための背景情報](#)」を確認してください。

- 連結ストライプを作成するには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「アクション」、「ボリュームを作成」の順に選択したあと、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
- コマンド行から既存のストライプを連結するには、次の形式の `metattach` コマンドを使用します。

```
# metattach volume-name component-names
```

`volume-name` 拡張するボリュームの名前を指定します。

`component-names` 使用されるコンポーネントの名前を指定します。複数のコンポーネントが使用される場合は、各コンポーネントをスペースで区切ります。

詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。

例 9-6 1つのスライスを接続することによる連結ストライプの作成

```
# metattach d2 c1t2d0s2
d2: components are attached
```

この例は、既存のストライプ `d2` にスライスを接続する方法を示しています。システムは、スライスが接続されていることを確認します。

例 9-7 複数のスライスを追加することによる連結ストライプの作成

```
# metattach d25 c1t2d0s2 c1t2d1s2 c1t2d3s2
d25: components are attached
```

この例では、既存の3方向のストライプ `d25` を取得し、それに別の3方向のストライプを連結します。接続されるスライスのインタレース値が指定されていないため、このストライプは、`d25` に対して構成されているインタレース値を継承します。システムは、ボリュームが設定されていることを確認します。

参照 UFS ファイルシステムの場合は、ボリュームに対して `growfs` コマンドを実行します。247 ページの「[ファイルシステムを拡張する方法](#)」を参照してください。

データベースなどの一部のアプリケーションは、ファイルシステムを使用しません。データベースなどのアプリケーションは raw ボリュームを使用するため、このボリュームを認識するか、または追加された容量を拡張するための独自の方法を備えている必要があります。

新しく作成された連結ストライプをファイルシステム用に準備するには、『[Oracle Solaris の管理: デバイスとファイルシステム](#)』の第 15 章「[ファイルシステムの作成およびマウント \(タスク\)](#)」を参照してください。

RAID-0 ボリュームの削除

▼ RAID-0 ボリュームを削除する方法

- 1 すべてのデータの現在のバックアップがあること、および **root** 権限を持っていることを確認します。

- 2 このボリュームが必要なくなっていることを確認します。

ストライプまたは連結を削除し、削除されたボリュームに含まれていたスライスを再利用すると、そのボリューム上のすべてのデータがシステムから削除されます。

- 3 必要に応じて、ファイルシステムをアンマウントします。

```
# umount /filesystem
```

- 4 ボリュームを削除するには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「編集」、「削除」の順に選択したあと、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metaclear` コマンドを使用して、ボリュームを削除します。

```
metaclear volume-name
```

詳細は、次の例および `metaclear(1M)` のマニュアルページを参照してください。

例 9-8 連結の削除

```
# umount d8
# metaclear d8
d8: Concat/Stripe is cleared
    (Edit the /etc/vfstab file)
```

この例は、マウントされたファイルシステムも含まれている連結 `d8` の削除を示しています。ボリュームを削除するには、その前にファイルシステムをアンマウントする必要があります。連結が削除されるという確認メッセージが表示されます。`/etc/vfstab` ファイル内にこのボリュームのエントリが存在する場合は、そのエントリを削除します。存在しないボリューム上にファイルシステムをマウントするよう指示してシステムを混乱させることがないようにしてください。

RAID-1 (ミラー) ボリューム (概要)

この章では、ミラーとサブミラーに関する基本的な Solaris Volume Manager の概念について説明します。関連タスクを実行する詳細については、第 11 章「RAID-1 (ミラー) ボリューム (タスク)」を参照してください。

この章では、次の内容について説明します。

- 99 ページの「RAID-1 (ミラー) ボリュームの概要」
- 102 ページの「RAID-1 ボリューム (ミラー) の再同期」
- 104 ページの「RAID-1 ボリュームの作成と保守」
- 109 ページの「RAID-1 ボリュームでシングルユーザーモードでブートする影響」
- 110 ページの「シナリオ - RAID-1 ボリューム (ミラー)」

RAID-1 (ミラー) ボリュームの概要

RAID-1 ボリュームまたは「ミラー」は、RAID-0 (ストライプまたは連結) ボリュームにデータの同一のコピーを維持しているボリュームです。ミラー化されている RAID-0 ボリュームは「サブミラー」と呼ばれます。ミラー化するには、多くのディスクが必要です。ミラー化するデータの量の少なくとも 2 倍のディスク領域が必要です。Solaris Volume Manager はすべてのサブミラーに書き込む必要があるため、ミラー化では、書き込みリクエストでディスクへの書き込みにかかる時間も長くなる可能性があります。

ミラーを構成すると、ミラーは物理スライスとまったく同じように使用できます。

既存のファイルシステムを含め、どのようなファイルシステムでもミラー化できます。これらのファイルシステムはルート (/)、swap、および /usr です。ミラーは、データベースなど、どのようなアプリケーションにも使用できます。

ヒント-データの安全性と可用性を維持するために、ミラーと Solaris Volume Manager のホットスペア機能を使用します。ホットスペアについては、[第 16 章「ホットスペアプール\(概要\)」](#)と[第 17 章「ホットスペアプール\(タスク\)」](#)を参照してください。

サブミラーの概要

ミラーは、サブミラーと呼ばれる 1 つまたは複数の RAID-0 ボリューム(ストライプまたは連結)から構成されます。

ミラーは最大 4 つのサブミラーで構成できます。ただし、2 方向のミラーは通常ほとんどのアプリケーションに十分なデータ冗長性を提供し、ディスクドライブコストの点で安価になります。3 つ目のサブミラーを構成すると、オンラインでバックアップをとることができます。この場合、バックアップのために 1 つのサブミラーがオフラインになっていても、データの冗長性は失われません。

サブミラーを「オフライン」にすると、ミラーはサブミラーの読み取りと書き込みを停止します。この時点で、たとえばバックアップを実行するために、サブミラー自体にアクセスできます。ただし、サブミラーは読み取り専用状態です。サブミラーがオフラインの間、Solaris Volume Manager はミラーへのすべての書き込みを追跡します。サブミラーがオンラインに戻ると、サブミラーがオフライン中に書き込まれたミラーの部分(「再同期領域」)だけが再同期されます。サブミラーは、エラーのある物理デバイスをトラブルシューティングまたは修復するために、オフラインにすることもできます。

サブミラーはいつでもミラーと接続または切断することができますが、少なくとも 1 つのサブミラーが常時接続されている必要があります。

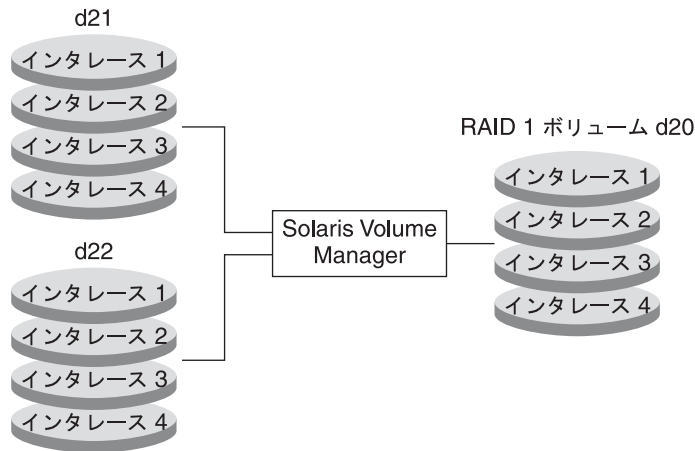
通常、ミラーは 1 つのサブミラーだけで作成します。ミラーの作成後に、2 番目のサブミラーを接続します。

シナリオ-RAID-1(ミラー)ボリューム

[図 10-1](#) に、ミラー d20 を示します。ミラーは 2 つのボリューム(サブミラー) d21 と d22 から構成されています。

Solaris Volume Manager は、複数の物理ディスクに、データの重複コピーを作成し、アプリケーションに対して 1 つの仮想ディスク(この例では d20)を提供します。すべてのディスクの書き込みが複製されます。ディスクの読み取りは基盤のいずれかのサブミラーから行なわれます。ミラー d20 の総容量は、最小のサブミラーのサイズになります(それらが等しいサイズでない場合)。

図 10-1 RAID-1(ミラー)の例



RAID-1+0 と RAID-0+1 の提供

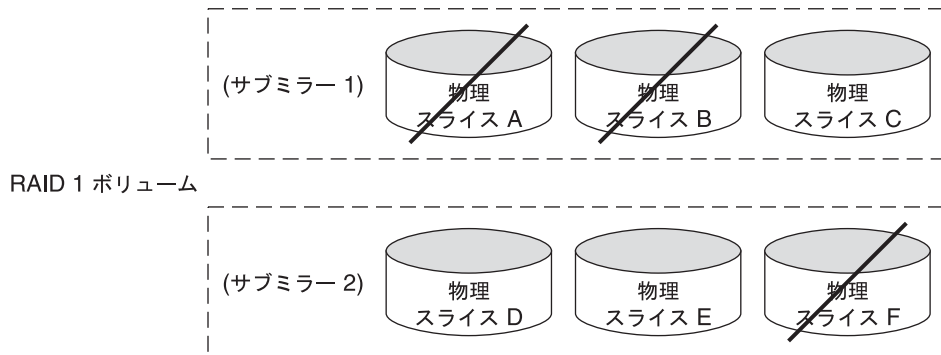
Solaris Volume Manager は RAID-1+0 と RAID-0+1 の両方の冗長性をサポートします。RAID-1+0 の冗長性は、ストライプ化されるミラーの構成から得られません。RAID-0+1 の冗長性は、ミラー化されるストライプの構成から得られません。Solaris Volume Manager インタフェースは、すべての RAID-1 デバイスが厳密に RAID-0+1 であるかのように見せます。ただし、Solaris Volume Manager は基盤のコンポーネントを認識し、可能な限りそれぞれを個別にミラー化します。

注 - Solaris Volume Manager は常に RAID-1+0 機能を提供できるとは限りません。ただし、両方のサブミラーが同一で、ディスクスライス(ソフトパーティションでなく)から構成されている場合、RAID-1+0 を提供できます。

3つのストライプ化されたスライスから構成される2方向ミラーのRAID-0+1実装を考慮します。Solaris Volume Managerがないと、1つのスライスの障害によって、ミラーの片側が機能しなくなることがあります。ホットスペアが使用されていないとすれば、2番目のスライスの障害によって、ミラーが機能しなくなる可能性があります。Solaris Volume Managerを使用すると、最大3つのスライスで障害が発生しても、ミラーが停止しません。ストライプ化された3つのスライスがそれぞれ、ミラーのもう半分の側の対応するスライスに対してミラー化されているため、ミラーは停止しません。

図 10-2 に、RAID-1 ボリュームでスライスの損失が発生しても、RAID-1+0 の実装によってデータの損失を防ぐしくみを示します。

図 10-2 RAID-1+0 の例



RAID-1 ボリュームは2つのサブミラーから構成されています。各サブミラーは、同じインターレース値を持つ3つの同一の物理ディスクから構成されています。3つのディスク A、B、Fの障害に耐えます。ミラーの論理ブロックの範囲全体が、少なくとも1つの正常なディスク上にまだ含まれています。ボリュームのすべてのデータが使用できます。

ただし、ディスク A と D で障害が発生した場合、ミラーのデータの一部がどのディスクでも使用できなくなります。これらの論理ブロックへのアクセスは失敗します。ただし、データを使用できるミラーの部分へのアクセスはまだ成功します。この状況では、ミラーは不良ブロックが発見された単一のディスクのように機能します。破損した部分は使用できませんが、残りの部分は使用できます。

RAID-1 ボリューム(ミラー)の再同期

RAID-1 ボリューム(ミラー)の再同期は、いずれかのサブミラーに次のことが発生した場合に、1つのサブミラーから別のサブミラーにデータをコピーするプロセスです。

- サブミラーの障害
- システムのクラッシュ
- オフラインであったサブミラーがオンラインに復帰
- 新しいサブミラーが追加された

再同期が行なわれている間、ミラーはユーザーによって読み取りおよび書き込み可能です。

ミラーの再同期により、実行中の書き込みを除いて、すべてのサブミラーを同一のデータで維持することによって、正しいミラー動作を確保します。

注- ミラーの再同期は省略できません。ミラーの再同期は手動で開始する必要はありません。このプロセスは自動的に実行されます。

完全な再同期

新しいサブミラーがミラーに接続される(追加される)と、ミラーの他方のサブミラーからすべてのデータが新しく接続されたサブミラーに自動的に書き込まれます。ミラーの再同期が完了すると、新しいサブミラーが読み取り可能になります。サブミラーは切断されるまで、ミラーに接続されたままになります。

再同期の実行中にシステムがクラッシュした場合、システムのリポートが終了すると、再同期が再開されます。

最適化された再同期

システムの障害発生後のリポート時、またはオフラインだったサブミラーがオンラインに復帰した場合に、Solaris Volume Manager は「最適化されたミラーの再同期」を実行します。メタディスクドライバはサブミラー領域を追跡します。この機能により、メタディスクドライバは、障害発生後に、同期しなくなった可能性のあるサブミラーを知ることができます。最適化されたミラーの再同期は、同期していない領域に対してのみ実行されます。リポート時に、ミラーが再同期される順番を指定できます。ミラーの再同期を省略するには、サブミラーのパス番号を0に設定します。パス番号の変更に関するタスクについては、[例 11-16](#)を参照してください。



注意-0のパス番号は、読み取り専用として再マウント化されたミラーに対してのみ使用してください。

部分的な再同期

サブミラー内のスライスの交換に続いて、Solaris Volume Manager はデータの「部分的なミラー再同期」を実行します。Solaris Volume Manager は、他方のサブミラーの残りの正常なスライスから、交換したスライスにデータをコピーします。

RAID-1 ボリュームの作成と保守

このセクションでは、ミラーの作成で役立つガイドラインを提供します。このセクションでは、作成するミラーのパフォーマンスに関するガイドラインも提供します。

RAID-1 ボリュームの構成のガイドライン

- ミラーを作成する前に、ミラーを構成する RAID-0 (ストライプまたは連結) ボリュームを作成します。
- ミラーを作成する場合、まず 1 方向のミラーを作成し、次に 2 番目のサブミラーを接続します。この方法により、再同期処理を開始します。さらに、この方法によりデータが破損しなくなります。1 方向のミラーは、将来の 2 方向または多方向のミラーとして使用するために作成することもできます。
- 1 つのコマンドで、1 方向ミラーから 2 方向ミラー、3 方向ミラー、4 方向ミラーを作成できます。1 つのコマンドで、すべてのサブミラーを作成することによって、作成プロセスを高速化できます。このプロセスは、既存のデータをミラー化しない場合およびすべてのサブミラー上のデータを破棄してもかまわない場合にのみ使用してください。
- スライス上に構築されている既存のファイルシステムから RAID-1 ボリュームを作成できます。プライマリ RAID-0 ボリューム (サブミラー) には、1 つのスライスのみを含めることができます。ルートまたは他のシステムクリティカルなファイルシステムをミラー化する場合、すべてのサブミラーが 1 つだけのスライスで構成されている必要があります。
- `swap-l` コマンドを使用して、すべての `swap` デバイスをチェックします。 `swap` として指定された各スライスは、残りのスワップスライスから独立してミラー化される必要があります。
- Solaris 管理コンソール内の拡張ストレージツールはミラー化されていないルート (/)、/opt、/usr、swap をサポートしません。実際に、ツールでは、システムの実行中にアンマウントできないファイルシステムのミラー化解除をサポートしていません。代わりに、これらのファイルシステムにはコマンド行プロシージャを使用します。
- 同じサイズのサブミラーを使用します。サイズが異なるサブミラーを使用すると、ディスク領域が無駄になります。
- ミラー内は、同じように構成されたサブミラーのみを使用します。特に、ラベルのないサブミラーでミラーを作成した場合、ディスクラベルを含むサブミラーを接続できなくなります。
- 最初に接続されたサブミラーがシリンダー 0 から開始していないミラー化されたファイルシステムを使用できます。接続するすべての追加のサブミラーもシリンダー 0 から始まっていない必要があります。この状況で始まっているサブミラーを接続しようとする、次のエラーメッセージが表示されます。

can't attach labeled submirror to an unlabeled mirror

特定のミラー内で使用することを意図されたサブミラーはすべてがシリンダー 0 から始まっているか、またはすべてがシリンダー 0 から始まっていない必要があります。

開始シリンダーは、すべてのサブミラーで同じである必要はありません。ただし、すべてのサブミラーにシリンダー 0 が含まれるか、または含まれていない必要があります。

- ミラーを作成する前に、状態データベースの複製を追加して、ミラーのパフォーマンスを向上できます。原則として、システムに追加するミラーごとに、2つの複製を追加します。Solaris Volume Manager はこれらの追加の複製を使用して、最適化された再同期を実行するために使用するダーティレーションログ (DRL) を保存します。適切な数の複製を提供することで、RAID-1 ボリュームのパフォーマンスに対する I/O の影響を最小にすることができます。同じディスクまたはコントローラ上の少なくとも2つの複製を、複製が記録するミラーとして使用することも全体的なパフォーマンスの向上に役立ちます。
- 直接マウントできるのはミラーデバイスだけです。サブミラーがオフラインで、読み取り専用でマウントする場合を除き、サブミラーを直接マウントしないでください。サブミラーの一部であるスライスを実装しないでください。このプロセスにより、データが破棄され、システムがクラッシュする可能性があります。

RAID-1 ボリュームのパフォーマンスのガイドライン

- サブミラーのスライスは、異なるディスクとコントローラに配置します。同じミラーの2つ以上のサブミラーのスライスを同じディスクに置くと、データの保護機能が大幅に低下します。同様に、サブミラーは、別個のコントローラに配置します。これは、コントローラやそのケーブルでは、ディスクよりも障害が発生する確率が高いためです。これにより、ミラーのパフォーマンスも向上します。
- 1つのミラーでは、同じタイプのディスクとコントローラを使用します。特に、古い SCSI ストレージデバイスでは、ディスクやコントローラのパフォーマンスがモデルやブランドによって大幅に異なることがあります。パフォーマンスレベルが異なるディスクやコントローラを1つのミラーで使用すると、パフォーマンスが大幅に低下することがあります。
- ミラー化すると読み取りパフォーマンスは向上する可能性がありますが、書き込みパフォーマンスは常に低下します。ミラー化によって読み取りパフォーマンスが向上するのは、スレッド化または非同期 I/O 状況の場合だけです。シングルスレッドによるボリュームの読み取りでは、パフォーマンスの向上は得られません。

- ミラーの読み取りポリシーを試してみて、パフォーマンスを向上させることができます。たとえば、デフォルトの読み取りモードは、ディスク間でラウンドロビン方式で交互に読み取ることです。このポリシーがデフォルトであるのは、ラウンドロビン方式がUFS マルチユーザー、マルチプロセッサアクティビティに最適に機能する傾向があるためです。

場合によっては、**geometric** 読み取りオプションによって、ヘッドの動きとアクセス時間を最小にすることで、パフォーマンスが向上することがあります。このオプションは、次の場合にもっとも効果的です。

- ディスクあたりにスライスが1つしかない
- 一度に1つのプロセスだけがスライスまたはファイルシステムを使用する
- I/O パターンの順次性が高いか、すべてのアクセスが読み取りの場合
- サービスを中断することなく、ミラーにサブミラーを接続できます。ミラーにサブミラーを接続して、2方向、3方向、4方向のミラーを作成します。
- サブミラーをオフラインにすると、ミラーがサブミラーとの読み取りや書き込みができなくなります。ただし、サブミラーのミラーとの論理的関連付けは維持されます。サブミラーがオフラインの間、Solaris Volume Manager はミラーへのすべての書き込みを追跡します。書き込みは、サブミラーがオンラインに戻ると書き込まれます。最適化された再同期の実行によって、Solaris Volume Manager はサブミラー全体ではなく、変更されたデータのみを再同期するだけで済みます。サブミラーを切断すると、ミラーへのその論理的関連付けも切断されます。通常、サブミラーをオフラインにするのは、保守を実行するためです。サブミラーを切断して取り外します。

RAID-1 ボリュームオプションについて

ミラーのパフォーマンスを最適にするために、次のオプションを使用できます。

- ミラーからの読み取りポリシー
- ミラーへの書き込みポリシー
- ミラーが再同期される順番 (パス番号)

最初にミラーを作成するときに、ミラーオプションを定義できます。ミラーを設定して、実行した後に、ミラーオプションを変更することもできます。これらのオプションの変更に関するタスクについては、[142 ページの「RAID-1 ボリュームオプションを変更する方法」](#)を参照してください。

RAID-1 ボリュームの読み取り/書き込みポリシー

Solaris Volume Manager では、RAID-1 ボリュームにさまざまな読み取り/書き込みポリシーを構成できます。読み取り/書き込みポリシーを適切に設定すると、特定の構成のパフォーマンスを向上できます。

表 10-1 RAID-1 ボリュームの読み取りポリシー

読み取りポリシー	説明
ラウンドロビン (デフォルト)	サブミラー間で負荷の分散を試みます。ミラー内のすべてのサブミラーから、すべての読み取りがラウンドロビン順 (1 つずつ順番に) で行なわれます。
ジオメトリック	論理ディスクブロックアドレスに基づいて、読み取りをサブミラー間で分割できます。たとえば、2 方向サブミラーでは、ミラー上のディスク領域が、2 つの等しいサイズの論理アドレス範囲に分割されます。1 つのサブミラーからの読み取りは、論理範囲の半分に制限されます。他方のサブミラーからの読み取りは、もう半分に制限されます。ジオメトリック読み取りポリシーでは、読み取りに必要なシーク時間を効率的に短縮できます。この読み取りポリシーによって得られるパフォーマンスは、システム I/O の負荷とアプリケーションのアクセスパターンによって異なります。
先頭のデバイスから読み取る	すべての読み取りを最初のサブミラーに送ります。このポリシーは、最初のサブミラーを構成しているデバイスが 2 番目のサブミラーのデバイスよりかなり高速である場合にのみ使用すべきです。

表 10-2 RAID-1 ボリュームの書き込みポリシー

書き込みポリシー	説明
並列 (デフォルト)	ミラーへの書き込みは複製され、すべてのサブミラーに同時に送られます。
順次	サブミラーへの書き込みは順次実行されます (つまり、最初のサブミラーへの書き込みが終わってから、2 番目のサブミラーへの書き込みが開始されます)。このポリシーでは、1 つのサブミラーへの書き込みが完了してから、次のサブミラーへの書き込みが開始されなければならないことを指定します。このポリシーは、電源の障害などによってサブミラーが読み取り不能になった場合に指定します。

パス番号

パス番号 (0 から 9 の範囲の数字) は、システムのリブート時に特定のミラーを再同期する順番を決定します。デフォルトのパス番号は 1 です。パス番号の小さいミラーが先に再同期されます。0 を使用すると、ミラーの再同期はスキップされません。パス番号 0 は、読み取り専用としてマウントされているミラーに対してのみ使用します。同じパス番号をもつミラーは同時に再同期されます。

保守作業を判断するためのサブミラーのステータスについて

Solaris Volume Manager の `metastat` コマンドは、RAID 1 ボリュームおよびサブミラーに関するステータス情報を報告します。ステータス情報は、RAID-1 ボリューム

への保守作業が必要かどうかを判断するために役立ちます。次の表に、RAID-1 ボリュームに対して、`metastat` コマンドを実行した場合に表示されるサブミラーの状態を説明します。

表 10-3 サブミラーの状態

状態	意味
正常	サブミラーはエラーがなく、正常に機能しています。
再同期中	サブミラーは再同期中です。エラーが発生して、修正され、サブミラーがオンラインに戻ったばかりか、新しいサブミラーが追加されました。
保守が必要	サブミラーのスライスで I/O エラーまたはオープンエラーが発生しました。サブミラー内のこのスライスのすべての読み取りと書き込みが中断されました。

さらに、サブミラー内の各スライスについて、`metastat` コマンドは次の情報を表示します。

デバイス	ストライプ内のスライスのデバイス名を示します
開始ブロック	スライスが開始するブロックを示します
Dbase	スライスに状態データベースの複製が格納されているかどうかを示します
状態	スライスの状態を示します
ホットスペア	スライスが障害のあるスライスのホットスペアとして使用されていることを示します。

サブミラーの状態は、サブミラーのステータスの一般的な情報のみを提供します。ミラーのエラーのトラブルシューティングを行なっている場合、スライスの状態は確認すべきもっとも重要な情報と考えられます。サブミラーで「保守が必要」状態が報告された場合、詳細について、スライスの状態を参照する必要があります。

スライスが「保守」状態にあるか、「最後にエラー」状態にあるかによって、異なる復元作業を実行します。「保守」状態のスライスしかない場合は、それらは任意の順番で修復できます。「保守」状態と「最後にエラー」状態の両方のスライスがある場合は、「保守」状態のスライスを先に修正する必要があります。「保守」状態のスライスを修正したら、次に、「最後にエラー」状態のスライスを修正します。詳細については、248 ページの「RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要」を参照してください。

次の表に、サブミラーのスライスの状態と、可能な対策を説明します。

表 10-4 サブミラーのスライスの状態

状態	意味	アクション
正常	スライスはエラーがなく、正常に機能しています。	なし。
再同期中	スライスは再同期中です。エラーが発生して、修正され、サブミラーがオンラインに戻ったばかりか、新しいサブミラーが追加されました。	必要に応じて、再同期が完了するまでサブミラーのステータスをモニターします。
保守	スライスに I/O エラーまたはオープンエラーが発生しました。このコンポーネントのすべての読み取りと書き込みが中断されました。	障害の発生したスライスを有効にするか、交換します。139 ページの「サブミラー内のスライスを有効にする方法」または 144 ページの「サブミラー内のスライスを置換する方法」を参照してください。metastat コマンドは、invoke 復元メッセージと metareplace コマンドによって実行する適切な対策を表示します。metareplace -e コマンドも使用できます。
最後にエラー	スライスに I/O エラーまたはオープンエラーが発生しました。ただし、データは別のスライスの障害のため、どこにも複製されません。I/O は、引き続きスライスで実行されます。I/O エラーが発生した場合、ミラーの I/O が失敗します。	まず、「保守」状態のスライスを有効にするか、交換します。139 ページの「サブミラー内のスライスを有効にする方法」または 144 ページの「サブミラー内のスライスを置換する方法」を参照してください。通常、このエラーによって一部のデータの損失が発生するため、ミラーの修正後に、ミラーを検証してください。ファイルシステムの場合、fsck コマンドを使用して、データをチェックします。アプリケーションやデータベースには、それぞれ固有のデバイスの検証方法があるはずです。

RAID-1 ボリュームでシングルユーザーモードでブートする影響

場合により、ルート (/)、/usr、および swap などのいわゆる「ブート」ファイルシステムのミラーのあるシステムをシングルユーザーモードでブートする (boot -s コマンドを使用して) 必要がある場合があります。この場合、metastat コマンドで表示した場合に、これらのミラーとおそらくシステム上のすべてのミラーが、「保守が必要」状態で表示されます。さらに、これらのスライスに対して書き込みが行なわれた場合に、metastat コマンドによって、ミラー上のダーティリージョンが増加していることが示されます。

この状況は、危険であるように見えます。ただし、通常、ブート中にミラーを再同期するために実行する `metasync -r` コマンドは、システムがシングルユーザーモードでブートすると中断されます。システムをリブートすると、`metasync -r` コマンドが実行され、すべてのミラーが再同期されます。

この状況が問題になる場合は、手動で `metasync -r` コマンドを実行してください。

シナリオ-RAID-1 ボリューム(ミラー)

RAID-1 ボリュームは、冗長ボリュームを構築する手段を提供します。そのため、基盤の RAID-0 ボリュームのいずれかの部分的または完全な障害が発生した場合も、データの損失やファイルシステムへのアクセスの中断が発生しません。次の例では、[第5章「Solaris Volume Managerの構成と使用\(シナリオ\)」](#)で説明され、[88ページの「シナリオ-RAID-0 ボリューム」](#)に続いているシナリオを利用して、RAID-1 ボリュームで冗長ストレージを提供する方法について説明します。

[88ページの「シナリオ-RAID-0 ボリューム」](#)で説明したように、サンプルシステムには、2つの RAID-0 ボリュームがあります。各ボリュームのサイズは約 27G バイトで、3つのディスクにまたがっています。RAID-1 ボリュームを作成して、これらの2つの RAID-0 ボリュームをミラー化することで、完全な冗長ストレージ領域により、弾性のあるデータストレージを提供できます。

この RAID-1 ボリューム内では、いずれかのディスクコントローラの障害によって、ボリュームへのアクセスが中断されません。同様に、最大3つの個別のディスクの障害もアクセスの中断なく耐えることができます。

アクセスを中断させる可能性のある問題に対する保護を追加するには、[第16章「ホットスペアプール\(概要\)」](#)で説明するように、ホットスペアを使用します。特に、[189ページの「ホットスペアのしくみ」](#)を参照してください。

RAID-1 (ミラー) ボリューム (タスク)

この章では、RAID-1 ボリュームに関連する Solaris Volume Manager のタスクを実行する方法について説明します。関連する概念については、第 10 章「RAID-1 (ミラー) ボリューム (概要)」を参照してください。

RAID-1 ボリューム (タスクマップ)

次のタスクマップは、Solaris Volume Manager の RAID-1 ボリュームを管理するために必要な手順を示します。

タスク	説明	参照先
未使用のスライスからミラーを作成する	Solaris Volume Manager の GUI または <code>metainit</code> コマンドを使用して、未使用のスライスからミラーを作成します。	113 ページの「未使用のスライスから RAID-1 ボリュームを作成する方法」
既存のファイルシステムからミラーを作成する	Solaris Volume Manager の GUI または <code>metainit</code> コマンドを使用して、既存のファイルシステムからミラーを作成します。	115 ページの「ファイルシステムから RAID-1 ボリュームを作成する方法」
ルート (/) ファイルシステムからミラーを作成する	Solaris Volume Manager の GUI または <code>metainit</code> コマンドを使用して、ルート (/) ファイルシステムからミラーを作成します。	120 ページの「SPARC: ルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法」 129 ページの「x86: DCA を使用してルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法」
サブミラーを接続する	Solaris Volume Manager の GUI または <code>metattach</code> コマンドを使用して、サブミラーを接続します。	136 ページの「サブミラーを接続する方法」

タスク	説明	参照先
サブミラーを切り離す	Solaris Volume Manager の GUI または <code>metadetach</code> コマンドを使用して、サブミラーを切り離します。	137 ページの「サブミラーを切り離す方法」
サブミラーをオンラインまたはオフラインにする	Solaris Volume Manager の GUI または <code>metaonline</code> コマンドを使用して、サブミラーをオンラインにします。Solaris Volume Manager の GUI または <code>metaoffline</code> コマンドを使用して、サブミラーをオフラインにします。	138 ページの「サブミラーをオフラインおよびオンラインにする方法」
サブミラー内のスライスを有効にする	Solaris Volume Manager の GUI または <code>metareplace</code> コマンドを使用して、サブミラー内のスライスを有効にします。	139 ページの「サブミラー内のスライスを有効にする方法」
ミラーのステータスをチェックする	Solaris Volume Manager の GUI または <code>metastat</code> コマンドを使用して、RAID-1 ボリュームのステータスをチェックします。	140 ページの「ミラーとサブミラーのステータスを表示する方法」
ミラーオプションを変更する	Solaris Volume Manager の GUI または <code>metaparam</code> コマンドを使用して、特定の RAID-1 ボリュームのオプションを変更します。	142 ページの「RAID-1 ボリュームオプションを変更する方法」
ミラーを拡張する	Solaris Volume Manager の GUI または <code>metattach</code> コマンドを使用して、ミラーの容量を拡張します。	143 ページの「RAID-1 ボリュームを拡張する方法」
サブミラー内のスライスを置き換える	Solaris Volume Manager の GUI または <code>metareplace</code> コマンドを使用して、サブミラー内のスライスを置き換えます。	144 ページの「サブミラー内のスライスを置換する方法」
サブミラーを置き換える	Solaris Volume Manager の GUI または <code>metattach</code> コマンドを使用して、サブミラーを置き換えます。	146 ページの「サブミラーを置換する方法」
ミラーを削除する(ミラー化を解除する)	Solaris Volume Manager の GUI、 <code>metadetach</code> コマンド、または <code>metaclear</code> コマンドを使用して、ファイルシステムのミラー化を解除します。	148 ページの「ファイルシステムのミラー化を解除する方法」
アンマウントできないファイルシステムのミラーを削除する(ミラー化を解除する)	Solaris Volume Manager の GUI、 <code>metadetach</code> コマンド、または <code>metaclear</code> コマンドを使用して、アンマウントできないファイルシステムのミラー化を解除します。	150 ページの「アンマウントできないファイルシステムのミラー化を解除する方法」

タスク	説明	参照先
ミラーを使用してバックアップを実行する	Solaris Volume Manager の GUI、 <code>metaonline</code> コマンド、または <code>metaoffline</code> コマンドを使用して、ミラーによるバックアップを実行します。	153 ページの「RAID-1 ボリュームのオンラインバックアップを実行する方法」

RAID-1 ボリュームの作成

▼ 未使用のスライスから RAID-1 ボリュームを作成する方法

この手順では、2 面ミラーを作成する方法を示します。3 面ミラーまたは 4 面ミラーを作成する場合も、同じ手順を使用します。

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 104 ページの「RAID-1 ボリュームの作成と保守」を確認してください。

- 1 ストライプまたは連結を 2 つ作成します。これらのコンポーネントはサブミラーになります。
90 ページの「RAID-0 (ストライプ) ボリュームを作成する方法」または 92 ページの「RAID-0 (連結) ボリュームを作成する方法」を参照してください。
- 2 ミラーを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択し、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用して 1 面ミラーを作成します。

```
# metainit volume-name -m submirror-name
```

`volume-name` 作成するボリュームの名前を指定します

`-m` ミラーを作成することを指定します

`submirror-name` ミラーの最初のサブミラーになるコンポーネントの名前を指定します

詳細は、次の例と `metainit(1m)` のマニュアルページを参照してください。
- 3 2 番目のサブミラーを追加するには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、変更するミラーを選択します。「アクション」、「プロパティ」、「サブミラー」の順に選択します。画面の指示に従ってサブミラーを接続します。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metattach` コマンドを使用します。

```
# metattach volume-name submirror-name
volume-name      サブミラーを追加する RAID-1 ボリュームの名前を指定します
submirror-name   ミラーに接続された 2 番目のサブミラーになるコンポーネント
                  の名前を指定します
```

詳細は、次の例と `metattach(1M)` のマニュアルページを参照してください。

例 11-1 2 面ミラーを作成する

```
# metainit d51 1 1 c0t0d0s2
d51: Concat/Stripe is setup
# metainit d52 1 1 c1t0d0s2
d52: Concat/Stripe is setup
# metainit d50 -m d51
d50: Mirror is setup
# metattach d50 d52
d50: Submirror d52 is attached
```

この例では、2 面ミラー `d50` を作成する方法を示します。 `metainit` コマンドは、RAID-0 ボリュームである 2 つのサブミラー (`d51` と `d52`) を作成します。 `metainit -m` コマンドは、`d51` RAID-0 ボリュームから 1 面ミラーを作成します。 `metattach` コマンドは、`d52` を接続して 2 面ミラーを作成し、再同期を実行させます。接続されたサブミラー上のデータは、再同期の際にもう一方のサブミラーによって上書きされます。

例 11-2 再同期なしで 2 面ミラーを作成する

```
# metainit d51 1 1 c0t0d0s2
d51: Concat/Stripe is setup
# metainit d52 1 1 c1t0d0s2
d52: Concat/Stripe is setup
# metainit d50 -m d51 d52
metainit: d50: WARNING: This form of metainit is not recommended.
The submirrors may not have the same data.
Please see ERRORS in metainit(1M) for additional information.
d50: Mirror is setup
```

この例では、2 面ミラー `d50` の作成を示します。 `metainit` コマンドは、RAID-0 ボリュームである 2 つのサブミラー (`d51` と `d52`) を作成します。その後、両方のサブミラーで `metainit -m` コマンドを実行して、ミラーを作成します。 `metattach` コマンドではなく `metainit` コマンドを使用してミラーを作成すると、再同期処理は実行され

ません。結果として、ミラーの両側が同一であり区別なく使用できると Solaris Volume Manager でみなされた場合、データが破壊される可能性があります。

参照 ファイルシステム用に新しく作成したミラーを準備するには、『Oracle Solaris の管理: デバイスとファイルシステム』の第 15 章「ファイルシステムの作成およびマウント (タスク)」を参照してください。データベースなどの一部のアプリケーションは、ファイルシステムを使用しません。これらのアプリケーションは、代わりに raw デバイスを使用します。このアプリケーションは、raw デバイスにアクセスするための独自の方法を備えている必要があります。

▼ ファイルシステムから RAID-1 ボリュームを作成する方法

既存のファイルシステムをミラー化するには、次の手順を使用します。ファイルシステムをアンマウントできる場合は、リブートなしで手順全体を完了することができます。/usr や /swap など、アンマウントできないファイルシステムの場合、手順を完了するにはシステムをリブートする必要があります。

スライス上に構築された既存のファイルシステムから RAID-1 ボリュームを作成する場合は、その単一のスライスだけをプライマリ RAID-0 ボリューム (サブミラー) に含めることができます。システムにとって重要なファイルシステムをミラー化する場合は、すべてのサブミラーが単一のスライスだけで構成されている必要があります。

ルート (/) ファイルシステムのミラー化に関連する手順については、120 ページの「SPARC: ルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法」および 129 ページの「x86: DCA を使用してルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法」を参照してください。

この手順で使用している例では、既存のスライスは `c1t0d0s0` です。2 番目のスライス `c1t1d0s0` はミラーの後半として使用できます。サブミラーはそれぞれ `d1` と `d2`、ミラーは `d0` です。



注意 - 必ず、`metainit` コマンドで 1 面ミラーを作成してから、`metattach` コマンドで追加のサブミラーを接続してください。`metattach` コマンドを使用しないと、再同期処理は実行されません。結果として、ミラーの両側が同一であり区別なく使用できると Solaris Volume Manager でみなされた場合、データが破壊される可能性があります。

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 104 ページの「RAID-1 ボリュームの作成と保守」を確認してください。

- 1 ミラー化する既存のファイルシステムが含まれているスライスを特定します。この例ではスライス `c1t0d0s0` を使用します。
- 2 次のいずれかの方法を使用して、前の手順のスライスに新しい RAID-0 ボリュームを作成します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

`-f` 強制的にコマンドを続行します。スライスにマウントされたファイルシステムが含まれている場合は、このオプションを使用する必要があります。

`volume-name` 作成するボリュームの名前を指定します。ボリュームの命名については、47 ページの「ボリューム名」を参照してください。

`number-of-stripes` 作成するストライプの数を指定します。

`components-per-stripe` 各ストライプに含めるコンポーネントの数を指定します。

`component-names` 使用されるコンポーネントの名前を指定します。この例ではルートスライス `c0t0d0s0` を使用します。

- 3 未使用のスライス(この例では `c1t1d0s0`)に、2 番目のサブミラーとして機能する 2 番目の RAID-0 ボリューム(連結)を作成します。2 番目のサブミラーは、最初のサブミラー以上のサイズでなければなりません。次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択し、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

注 - オプションの説明については、手順 2 を参照してください。

- 4 次のいずれかの方法を使用して、1 面ミラーを作成します。
 - Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。

- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name -m submirror-name
```

`volume-name` 作成するボリュームの名前を指定します。

`-m` ミラーを作成することを指定します。

`submirror-name` ミラーの最初のサブミラーになるコンポーネントの名前を指定します。この例では、ルートスライスを含む RAID-0 ボリュームです。

詳細は、[metainit\(1m\)](#) のマニュアルページを参照してください。



注意-既存のファイルシステムからミラーを作成する場合は、データの破壊を防ぐために、次の2つの手順に忠実に従ってください。

- 5 `/etc/vfstab` ファイルを編集して、ファイルシステムのマウント手順がブロックデバイスではなくミラーを参照するようにします。`/etc/vfstab` ファイルの詳細は、『[Oracle Solarisの管理: デバイスとファイルシステム](#)』の「[ファイルシステムのマウントおよびマウント解除の概要](#)」を参照してください。

たとえば、`/etc/vfstab` ファイルにファイルシステムの次のエントリが含まれている場合は、

```
/dev/dsk/slice /dev/rdisk/slice /var ufs 2 yes -
```

そのエントリを次のように変更します。

```
/dev/md/dsk/mirror-name /dev/md/rdsk/mirror-name /var ufs 2 yes -
```

- 6 次のいずれかの方法に従って、新しくミラー化したファイルシステムを再びマウントします。
 - アンマウント可能なファイルシステムをミラー化している場合は、ファイルシステムをアンマウントしてから再びマウントします。

```
# umount /filesystem
```

```
# mount /filesystem
```

- アンマウントできないファイルシステムをミラー化している場合は、システムをリブートします。

```
# reboot
```

- 7 次の形式の `metattach` コマンドを使用して2番目のサブミラーを接続します。

```
# metattach volume-name submirror-name
```

`volume-name` サブミラーを追加する RAID-1 ボリュームの名前を指定します

`submirror-name` ミラーに接続された2番目のサブミラーになるコンポーネントの名前を指定します

詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。

例 11-3 アンマウント可能なファイルシステムから 2 面ミラーを作成する

```
# metainit -f d1 1 1 c1t0d0s0
d1: Concat/Stripe is setup
# metainit d2 1 1 c1t1d0s0
d2: Concat/Stripe is setup
# metainit d0 -m d1
d0: Mirror is setup
# umount /master
  (Edit the /etc/vfstab file so that the file system references the mirror)
# mount /master
# metattach d0 d2
d0: Submirror d2 is attached
```

この例では、`-f` オプションで最初の連結 `d1` を強制的に作成します。これには、`/dev/dsk/c1t0d0s0` にマウントされたファイルシステム `/master` が含まれています。2 番目の連結 `d2` を `/dev/dsk/c1t1d0s0` から作成します。このスライスは、`d1` 以上のサイズでなければなりません。)-`m` オプションを指定した `metainit` コマンドで、`d1` から 1 面ミラー `d0` を作成します。

次に、`/etc/vfstab` ファイルで、このファイルシステムのエントリがミラーを参照するように変更します。`/etc/vfstab` ファイルで、次の行は最初は次のようになっています。

```
/dev/dsk/c1t0d0s0 /dev/rdisk/c1t0d0s0 /var ufs 2 yes -
```

エントリを次のように変更します。

```
/dev/md/dsk/d0 /dev/md/rdisk/d0 /var ufs 2 yes -
```

最後に、ファイルシステムを再びマウントし、サブミラー `d2` をミラーに接続して、ミラーの再同期を実行させます。RAID-0 および RAID-1 ボリュームが設定され、サブミラー `d2` が接続されたことを示すメッセージが表示されます。

例 11-4 アンマウントできないファイルシステムから 2 面ミラーを作成する

```
# metainit -f d12 1 1 c0t3d0s6
d12: Concat/Stripe is setup
# metainit d22 1 1 c1t0d0s6
d22: Concat/Stripe is setup
# metainit d2 -m d12
d2: Mirror is setup
  (Edit the /etc/vfstab file so that /usr references the mirror)
# reboot
...
# metattach d2 d22
d2: Submirror d22 is attached
```

この例では、`/usr` ファイルシステムを含むスライスを使用して2面ミラーを作成します。`-f` オプションで最初の連結 `d12` を強制的に作成します。これには、`/dev/dsk/c0t3d0s6` にマウントされたファイルシステム `/usr` が含まれています。2番目の連結 `d22` を `/dev/dsk/c1t0d0s6` から作成します。このスライスは、`d12` 以上のサイズでなければなりません。`-m` オプションを指定した `metainit` コマンドで、`/usr` ファイルシステムを含む連結を使用して1面ミラー `d2` を作成します。次に、`/etc/vfstab` ファイルを編集して、`/usr` のエントリがミラーを参照するように変更する必要があります。

`/etc/vfstab` ファイルには、`/usr` ファイルシステムのエントリが次のように指定されています。

```
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /usr ufs 1 yes -
```

そのエントリを次のように変更します。

```
/dev/md/dsk/d2 /dev/md/rdisk/d2 /usr ufs 1 yes -
```

リポート後、2番目のサブミラー `d22` をミラーに接続して、ミラーの再同期を実行させます。

例 11-5 /swap 領域からミラーを作成する

```
# metainit -f d11 1 1 c0t0d0s1
d11: Concat/Stripe is setup
# metainit d21 1 1 c1t0d0s1
d21: Concat/Stripe is setup
# metainit d1 -m d11
d1: Mirror is setup
      (Edit the /etc/vfstab file so that swap references the mirror)
# reboot
...
# metattach d1 d21
d1: Submirror d21 is attached
```

この例では、`-f` オプションで最初の連結 `d11` を強制的に作成します。これには、`/dev/dsk/c0t0d0s1` にマウントされたファイルシステム `swap` が含まれています。2番目の連結 `d21` を `/dev/dsk/c1t0d0s1` から作成します。このスライスは、`d11` 以上のサイズでなければなりません。`-m` オプションを指定した `metainit` コマンドで、`swap` ファイルシステムを含む連結を使用して1面ミラー `d1` を作成します。次に、`/etc/vfstab` ファイルに `swap` のエントリがある場合は、ミラーを参照するように編集する必要があります。

`/etc/vfstab` ファイルには、`swap` 領域のエントリが次のように指定されています。

```
/dev/dsk/c0t0d0s1 - - swap - no -
```

そのエントリを次のように変更します。

```
/dev/md/dsk/d1 - - swap - no -
```

リブート後、2番目のサブミラー d21 をミラーに接続して、ミラーの再同期を実行させます。

swap 領域をミラー化した場合、クラッシュダンプを保存するには、`dumpadm` コマンドを使用してダンプデバイスをボリュームとして構成します。たとえば、スワップデバイスの名前が `/dev/md/dsk/d2` であれば、`dumpadm` コマンドを使用してこのデバイスをダンプデバイスとして設定します。

▼ SPARC: ルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法

SPARC プラットフォームでルート (/) ファイルシステムをミラー化するプロセスは、アンマウントできない他のファイルシステムをミラー化するプロセスと同様です。手順が異なるのは、`/etc/vfstab` ファイルを手動で編集する代わりに `metaroot` コマンドを実行する点です。ルート (/) ファイルシステムをミラー化するには、代替ブートデバイスのパスを記録する必要もあります。このデバイスは、サブミラーに障害が発生した場合にシステムをリブートします。

この手順で使用している例では、既存のスライスは `c1t0d0s0` です。2番目のスライス `c1t1d0s0` はミラーの後半として使用できます。サブミラーはそれぞれ `d1` と `d2`、ミラーは `d0` です。



注意 - 必ず、`metainit` コマンドで1面ミラーを作成してから、`metattach` コマンドで追加のサブミラーを接続してください。`metattach` コマンドを使用しないと、再同期処理は実行されません。結果として、ミラーの両側が同一であり区別なく使用できると Solaris Volume Manager でみなされた場合、データが破壊される可能性があります。

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 104 ページの「RAID-1 ボリュームの作成と保守」を確認してください。

- 1 ミラー化する既存のルート (/) ファイルシステムが含まれているスライスを特定します。この例ではスライス `c1t0d0s0` を使用します。
- 2 次のいずれかの方法を使用して、前の手順のスライスに新しい RAID-0 ボリュームを作成します。RAID-0 ボリュームに含めることができるのは、単一のスライスだけです。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

`-f` 強制的にコマンドを続行します。スライスにマウントされたファイルシステムが含まれている場合は、このオプションを使用する必要があります。

`volume-name` 作成するボリュームの名前を指定します。ボリュームの命名については、47 ページの「ボリューム名」を参照してください。

`number-of-stripes` 作成するストライプの数を指定します。

`components-per-stripe` 各ストライプに含めるコンポーネントの数を指定します。

`component-names` 使用されるコンポーネントの名前を指定します。この例ではルートスライス `c0t0d0s0` を使用します。

- 3 未使用のスライス(この例では `c1t1d0s0`)に、2 番目のサブミラーとして機能する 2 番目の RAID-0 ボリュームを作成します。セカンダリサブミラーは、最初のサブミラー以上のサイズでなければなりません。次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択し、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

注 - オプションの説明については、手順 2 を参照してください。

- 4 次のいずれかの方法を使用して、1 面ミラーを作成します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name -m submirror-name
```

`volume-name` 作成するボリュームの名前を指定します。

`-m` ミラーを作成することを指定します。

submirror-name ミラーの最初のサブミラーになるコンポーネントの名前を指定します。この例では、ルートスライスを含む RAID-0 ボリュームです。

- 新しくミラー化したファイルシステムを再びマウントします。**metaroot** *volume-name* コマンドを実行します (*volume-name* は、作成したミラーの名前で置き換えてください)。その後、システムをリブートします。

```
# metaroot volume-name
# reboot
```

詳細は、[metaroot\(1M\)](#) のマニュアルページを参照してください。

- 次の形式の **metattach** コマンドを使用して 2 番目のサブミラーを接続します。

```
# metattach volume-name submirror-name
```

volume-name サブミラーを追加する RAID-1 ボリュームの名前を指定します

submirror-name ミラーに接続された 2 番目のサブミラーになるコンポーネントの名前を指定します

詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。

- 代替ブートパスを記録します。

- 代替ルートデバイスのパスを調べます。ルート (/) ファイルシステムミラーに 2 番目のサブミラーとして接続されているスライスに対して **ls -l** コマンドを使用します。

```
# ls -l /dev/dsk/c1t1d0s0
lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/rdisk/c1t1d0s0 -> \
../devices/sbus@1,f8000000/esp@1,200000/sd@3,0:a
```

- /devices** ディレクトリに続く次の文字列 **/sbus@1,f8000000/esp@1,200000/sd@3,0:a** を記録しておきます。

注-システムが使用不可になる場合もあるため、この情報はシステム以外の場所にも書き留めるようにしてください。代替ブートデバイスからのブートの詳細については、[300 ページの「ブートの問題からの回復」](#)を参照してください。

- 文字列を編集してメジャー名(この場合は **sd**)を **disk** に変更し、**/sbus@1,f8000000/esp@1,200000/disk@3,0:a** とします。システムで IDE バスが使用されている場合、元のフルパスは次のようになっています。

```
$ ls -l /dev/dsk/c1t1d0s0
lrwxrwxrwx 1 root root 38 Mar 13 15:03 /dev/dsk/c0t0d0s0 -> \
../devices/pci@1f,0/ide@d/dad@0,0:a
```

メジャー名 **dad** を **disk** に変更すると、**/pci@1f,0/ide@d/disk@0,0:a** となります

- d. OpenBoot PROM の `nvalias` コマンドを使用して、セカンダリルート (/) ファイルシステムミラーの「バックアップルート」デバイス別名を定義します。例:

```
ok nvalias backup_root /sbus@1,f8000000/esp@1,200000/disk@3,0:a
```

- e. プライマリサブミラーとセカンダリサブミラーの両方を参照するように `boot-device` 別名を再定義し(サブミラーは指定された順に使用されます)、構成を保存します。

```
ok printenv boot-device
boot-device =          disk net
ok setenv boot-device disk backup_root net
boot-device =          disk backup_root net
ok nvstore
```

注-プライマリサブミラーに障害が発生した場合、システムは自動的にセカンダリサブミラーからブートします。自動ブートを使用せずに手動でブートする場合は、次のように入力します。

```
ok boot backup_root
```

例 11-6 SPARC: ルート (/) ファイルシステムからミラーを作成する

```
# metainit -f d1 1 1 c0t0d0s0
d1: Concat/Stripe is setup
# metainit d2 1 1 c0t1d0s0
d2: Concat/Stripe is setup
# metainit d0 -m d1
d0: Mirror is setup
# metaroot d0
# lockfs -fa
# reboot
...
# metattach d0 d2
d0: Submirror d2 is attached
# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root          88 Feb  8 15:51 /dev/rdsk/clt3d0s0 ->
../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a
# init 0
.
.
.
ok nvalias backup_root /pci@1f,0/pci@1,1/ide@3/disk@0,0:a
ok setenv boot-device disk backup_root net
ok nvstore
```

この例では、`-f` オプションで最初の RAID-0 ボリューム `d1` を強制的に作成します。これには、`/dev/dsk/c0t0d0s0` にマウントされたルート (/) ファイルシステムが含まれています。2 番目の連結 `d2` を `/dev/dsk/c0t1d0s0` から作成します。このスライスは、`d1` 以上のサイズでなければなりません。`-m` オプションを指定した `metainit` コマンドで、ルート (/) を含む連結を使用して 1 面ミラー `d0` を作成します。

次に、`metaroot` コマンドで `/etc/vfstab` および `/etc/system` ファイルを編集して、システムがボリューム上のルート (`/`) ファイルシステムからブートできるようにします。リブートの前に `lockfs -fa` コマンドを実行することをお勧めします。詳細は、[lockfs\(1M\)](#) のマニュアルページを参照してください。

システムをリブートする前に 2 番目のサブミラーを接続しないでください。`metaroot` コマンドの実行後、2 番目のサブミラーを接続する前にリブートする必要があります。

リブート後、サブミラー `d2` をミラーに接続して、ミラーの再同期を実行させます。連結とミラーが設定され、サブミラー `d2` が接続されたことを示すメッセージが表示されます。

あとでシステムを代替ルートデバイスからブートする必要がある場合に備え、ルート `raw` デバイスに対して `ls -l` コマンドを実行して、代替ルートデバイスのパスを調べます。

x86: ルート (`/`) ファイルシステムから RAID-1 ボリュームを作成する

Solaris 10 1/06 リリース以降、x86 ベースシステムのブートプロセスと構成に使用される Device Configuration Assistant (デバイス構成用補助、DCA) が GRand Unified Bootloader (GRUB) に置き換えられました。この機能の簡単な説明については、『Oracle Solaris の管理: 基本管理』の第 15 章「GRUB ベースのブート (参照情報)」を参照してください。

このセクションでは、ルート (`/`) ファイルシステムから RAID-1 ボリュームを作成する手順について説明します。Solaris 10 1/06 以降のリリースの OS を実行しているシステムでは、GRUB を使用する最初の手順に従ってください。それ以外の場合は、DCA を使用する 2 番目の手順を実行します。

x86 ベースシステムでルート (`/`) ファイルシステムをミラー化するプロセスは、SPARC システムでルートをミラー化するプロセスと同様です。ただし、x86 ベースシステムでは、BIOS と `fdisk` パーティションという複雑な階層があります。

この手順で使用している例では、既存のスライスは `c1t0d0s0` です。2 番目のスライス `c1t1d0s0` はミラーの後半として使用できます。サブミラーはそれぞれ `d1` と `d2`、ミラーは `d0` です。

注 - 手順を実行する前に、51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 104 ページの「RAID-1 ボリュームの作成と保守」を参照してください。

▼ x86: GRUB を使用してルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法

- 1 ミラーの 2 番目のディスクからシステムがブートできるように、BIOS ブートデバイスの順番を構成できることを確認します。

カーネルが起動される前は、システムは、x86 ベースシステム上のファームウェアインタフェースである、読み取り専用メモリー (ROM) の Basic Input/Output System (BIOS) によって制御されます。この BIOS は、SPARC ベースシステムのブート PROM に似ています。BIOS のタスクには次のようなものがあります。

- 起動機能を実行します。
- システムをブートする正しいデバイスを検出します。
- そのデバイスからマスターブートレコードをロードして、システムがセルフブートできるようにします。

通常は、BIOS を構成して、ブートレコードを探すデバイスの順番を選択できます。さらに、最近のほとんどの BIOS 実装では、セカンダリサブミラーへのフェイルオーバーが自動的に行われるようにデバイスを構成できます。システムの BIOS にこの機能がなく、プライマリサブミラーに障害が発生した場合は、システムのブート中に BIOS にアクセスし、セカンダリルートスライスからブートするようにシステムを再構成する必要があります。BIOS の設定を構成する方法については、BIOS のユーザーガイドを参照してください

ルートミラーを設定する前に、システム上の BIOS で、複数のディスクからブートできることを確認します。デバイスドライバによっては、システムの 1 つのディスクしか認識しないように構成されているものもあります。

- 2 **fdisk** パーティションがルートのミラー化をサポートするように構成されていることを確認します。

独立した x86 ブートパーティションが存在する場合は、ルート (/) ファイルシステムのミラー化中に問題が発生します。x86 ブートパーティションは Solaris **fdisk** パーティションの外部に存在しているため、Solaris Volume Manager ではミラー化できません。さらに、x86 ブートパーティションのコピーは 1 つしかないため、単一障害点にもなります。

Solaris 10 1/06 リリース以降の GRUB ベースのインストールプログラムでは、x86 ブートパーティションは自動的に作成されません。しかし、システムに x86 がすでに存在する場合、インストールプログラムはデフォルトでそのパーティションを保存します。

システムに独立した x86 ブートパーティションがあるかどうかを調べるには、`/etc/vfstab` ファイルを確認します。このファイルに次のようなエントリが含まれている場合、x86 ブートパーティションは存在しています。

```
/dev/dsk/c2t1d0p0:boot - /boot pcfs - no -
```

Solaris Volume Manager を使ってルート (/) ファイルシステムをミラー化する場合、このファイルシステムで単一の Solaris fdisk パーティションを使用する必要があります。したがって、システム内にすでに x86 ブートパーティションが存在する場合は、fdisk コマンドでこのパーティションを削除してから、Solaris ソフトウェアを再インストールしてください。再インストール時には、ブートパーティションは作成されません。

注 - Solaris Volume Manager でミラー化できるのは、Solaris fdisk パーティション内のスライスだけです。複数の fdisk パーティションが存在する場合、Solaris fdisk パーティションの外にあるデータを保護するには、別の方法を使用する必要があります。

- 3 マスターブートプログラムを使って、セカンダリサブミラーをブート可能にします。

- a. マスターブートプログラムを指定します。

```
# fdisk -b /usr/lib/fs/ufs/mboot /dev/rdisk/c1t1d0p0
```

次の画面が表示されます。

```
Total disk size is 31035 cylinders
Cylinder size is 1146 (512 byte) blocks
```

Partition	Status	Type	Cylinders		Length	%
			Start	End		
=====	=====	=====	=====	=====	=====	=====
1	Active	Solaris	1	31034	31034	100

SELECT ONE OF THE FOLLOWING:

1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Enter Selection:

- b. メニューから 5 番を選択して、Return キーを押します。

- 4 セカンダリディスクをブート可能にします。

```
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdisk/c1t1d0s0
```

installgrub の詳細は、[installgrub\(1M\)](#) のマニュアルページを参照してください。

- 5 ミラー化する既存のルート (/) ファイルシステムが含まれているスライスを特定します。

この例ではスライス c1t0d0s0 を使用します。

6 前の手順のスライスに新しい RAID-0 ボリュームを作成します。

RAID-0 ボリュームに含めることができるのは、単一のスライスだけです。次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

<code>-f</code>	強制的にコマンドを続行します。スライスにマウントされたファイルシステムが含まれている場合は、このオプションを使用する必要があります。
<code>volume-name</code>	作成するボリュームの名前を指定します。ボリュームの命名については、47 ページの「ボリューム名」を参照してください。
<code>number-of-stripes</code>	作成するストライプの数を指定します。
<code>components-per-stripe</code>	各ストライプに含めるコンポーネントの数を指定します。
<code>component-names</code>	使用されるコンポーネントの名前を指定します。この例ではルートスライス <code>c0t0d0s0</code> を使用します。

7 未使用のスライスに、2 番目のサブミラーとして機能する 2 番目の RAID-0 ボリューム (この例では `c1t1d0s0`) を作成します。

注-セカンダリサブミラーは、最初のサブミラー以上のサイズでなければなりません。また、2 番目のサブミラーとして使用するスライスには「root」というスライスタグを付ける必要があり、ルートスライスはスライス 0 である必要があります。

スライスタグフィールドの構成方法については、`format(1M)` のマニュアルページを参照してください。

次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択し、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes components-per-stripes component-names
```

注-オプションの説明については、手順6を参照してください。

8 次のいずれかの方法を使用して、1面ミラーを作成します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name -m submirror-name
```

`volume-name` 作成するボリュームの名前を指定します。

`-m` ミラーを作成することを指定します。

`submirror-name` ミラーの最初のサブミラーになるコンポーネントの名前を指定します。この例では、ルートスライスを含む RAID-0 ボリュームです。

9 新しくミラー化したファイルシステムを再びマウントしてから、システムをリブートします。

```
# metaroot volume-name
```

```
# reboot
```

詳細は、[metaroot\(1M\)](#) のマニュアルページを参照してください。

10 2番目のサブミラーを接続します。

```
# metattach volume-name submirror-name
```

`volume-name` サブミラーを追加する RAID-1 ボリュームの名前を指定します。

`submirror-name` ミラーに接続された2番目のサブミラーになるコンポーネントの名前を指定します。

詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。

11 `menu.lst` ファイルに代替ブートパスを定義します。

セカンダリサブミラーを持つディスクからシステムがブートできるようにするには、そのディスクを代替ブートデバイスとして認識するようにシステムを構成します。現在の `c1t1d0s0` の例では、代替パスは、2番目のディスクの最初の `fdisk` パーティションの、最初のスライス上にあります。したがって、`menu.lst` を次のように編集します。

```
title alternate boot
  root (hd1,0,a)
  kernel /boot/multiboot
  module /boot/x86.miniroot-safe
```

注 - menu.lst のエントリを適切に編集するには、GRUB のディスク命名規則について熟知する必要があります。詳細は、『Oracle Solaris の管理: 基本管理』の「GRUB を使用して x86 システムをブートする (タスクマップ)」を参照してください

menu.lst ファイルの編集後、システムは 2 番目のディスクにフェイルオーバーするように設定されます。プライマリディスクに障害が発生した場合、システムがセカンダリディスクからブートするように、ディスク番号が変更されます。



注意 - 場合によっては、BIOS の自動ディスク番号再割り当て機能が、使用できなくなったプライマリディスクからの回復に影響を及ぼすことがあります。ディスク番号再割り当て機能により、システムが強制的にセカンダリディスクからブートするようになると、プライマリディスクのブートアーカイブは無効になります。同じプライマリディスクがあとで使用可能になって、システムをブートした場合、デフォルトのプライマリディスクからシステムがブートするように、ディスク番号が再度切り替わります。しかし、この段階では、プライマリディスクのブートアーカイブは無効なままです。したがって、システムはまったくブートしない可能性があります。このような場合は、GRUB メニューから正しいエントリを選択して、システムが有効なブートアーカイブからブートするようにします。システムのブートプロセスが完了したら、通常どおり metadvice による保守を実行して、プライマリディスクとセカンダリディスクの両方を同期化し、再度プライマリディスクを有効なブートアーカイブにします。

▼ x86: DCA を使用してルート (/) ファイルシステムから RAID-1 ボリュームを作成する方法

- 1 ミラーの 2 番目のディスクからシステムがブートできるように、BIOS ブートデバイスの順番を構成できることを確認します。

カーネルが起動される前は、システムは、x86 ベースシステム上のファームウェアインタフェースである、読み取り専用メモリー (ROM) の Basic Input/Output System (BIOS) によって制御されます。この BIOS は、SPARC ベースシステムのブート PROM に似ています。ほかの起動機能に加えて、BIOS は正しいブートデバイスを見つけ、そのデバイスからマスターブートレコードを読み込んで、システムがブートできるようにします。通常は、BIOS を構成して、ブートレコードを探すデバイスの順番を選択できます。さらに、最近のほとんどの BIOS 実装では、セカンダリサブミラーへのフェイルオーバーが自動的に行われるようにデバイスを構成できます。システムにこの機能がなく、プライマリサブミラーに障害が発生した場合は、システムのブート中に BIOS にアクセスし、セカンダリルートスライスからブートするようにシステムを再構成する必要があります。BIOS の設定を構成する方法については、BIOS のユーザーガイドを参照してください。

システム上の DCA を使って、複数のディスクからブート可能であることを確認できます。デバイスドライバによっては、システムの1つのディスクしか認識しないように構成されているものもあります。

- 2 **fdisk** パーティションがルートのミラー化をサポートするように構成されていることを確認します。

x86 ベースシステムのもう一つの特長は、**fdisk** パーティションを使用することです。Solaris OS インストールプログラムのブートディスクパーティションのデフォルトのレイアウトでは、Solaris **fdisk** パーティションのほかに、「x86 ブートパーティション」と呼ばれる約 10M バイトの小さな **fdisk** パーティションが作成されます。

x86 ブートパーティションは、ルート (*/*) ファイルシステムをミラー化するときに問題になります。x86 ブートパーティションは Solaris **fdisk** パーティションの外部にあります。そのため、x86 ブートパーティションは Solaris Volume Manager ではミラー化できません。さらに、x86 ブートパーティションのコピーは1つしかないため、単一障害点にもなります。

Solaris OS に独立した x86 ブートパーティションがあるかどうかを調べることができます。x86 ブートパーティションは、`/etc/vfstab` ファイル内の次のようなエントリでマウントされます。

```
/dev/dsk/c2t1d0p0:boot - /boot pcfs - no -
```

独立した x86 ブートパーティションが存在しない場合、このエントリは `/etc/vfstab` ファイルに存在しません。

ルート (*/*) ファイルシステムをミラー化するためには、**fdisk** パーティションをカスタマイズして、x86 ブートパーティションを削除し、単一の Solaris **fdisk** パーティションを使用するようにします。Solaris Volume Manager のルートミラー化を使用する予定がある場合、システムのインストール時に、独立した x86 ブートパーティションを作成しないでください。すでにシステムがインストールされ、独立した x86 ブートパーティションが作成されている場合、**fdisk** コマンドを使用してその **fdisk** パーティションを削除し、システムをインストールし直します。インストール中に独立した x86 ブートパーティションを作成しないようにするには、インストールプロセス中にディスクパーティションをカスタマイズします。

注 - Solaris Volume Manager でミラー化できるのは、Solaris **fdisk** パーティション内のスライスだけです。複数の **fdisk** パーティションが存在する場合、Solaris **fdisk** パーティションの外にあるデータを保護するには、別の方法を使用する必要があります。

- 3 マスターブートプログラムを使って、セカンダリサブミラーをブート可能にします。

- a. **fdisk** コマンドを使用して、マスターブートプログラムを指定します。

```
# fdisk -b /usr/lib/fs/ufs/mboot /dev/rdisk/c1t1d0p0
```

次の画面が表示されます。

```
Total disk size is 31035 cylinders
Cylinder size is 1146 (512 byte) blocks
```

Partition	Status	Type	Cylinders		Length	%
			Start	End		
1	Active	Solaris	1	31034	31034	100

SELECT ONE OF THE FOLLOWING:

1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Enter Selection:

- b. メニューから 5 番を選択して、**Return** キーを押します。

- 4 セカンダリサブミラーをブート可能にするために、セカンダリサブミラーにブートブロックをインストールします。

この **fdisk** パーティションから Solaris OS をブートするには、セカンダリサブミラーがあるディスクにスライス 8 が必要です。このスライスには、パーティションブートレコード (pboot)、当該ディスクの Solaris VTOC、およびブートブロックが含まれます。この情報はディスクに固有なので、Solaris Volume Manager ではミラー化されません。しかし、プライマリディスクに障害が発生した場合にセカンダリディスクからブートできるようにするには、両方のディスクがブート可能であることを保証する必要があります。**installboot** コマンドを使って、2 番目のディスクを Solaris ブート可能ディスクとして設定します。詳細は、[installboot\(1M\)](#) のマニュアルページを参照してください。

このディスクのスライス 2 をデバイスとして指定する必要があり、このスライス 2 はディスク全体から構成されている必要があります。

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot \
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c1t1d0s2
```

- 5 ミラー化する既存のルート (/) ファイルシステムが含まれているスライスを特定します。この例ではスライス **c1t0d0s0** を使用します。

- 6 次のいずれかの方法を使用して、前の手順のスライスに新しい **RAID-0** ボリュームを作成します。RAID-0 ボリュームに含めることができるのは、単一のスライスだけです。
- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用します。

```
# metainit -f volume-name number-of-stripes components-per-stripe component-name
```

`-f` 強制的にコマンドを続行します。スライスにマウントされたファイルシステムが含まれている場合は、このオプションを使用する必要があります。

`volume-name` 作成するボリュームの名前を指定します。ボリュームの命名については、47 ページの「ボリューム名」を参照してください。

`number-of-stripes` 作成するストライプの数を指定します。

`components-per-stripe` 各ストライプに含めるコンポーネントの数を指定します。

`component-names` 使用されるコンポーネントの名前を指定します。この例ではルートスライス `c0t0d0s0` を使用します。

- 7 未使用のスライス(この例では `c1t1d0s0`)に、2 番目のサブミラーとして機能する 2 番目の **RAID-0** ボリュームを作成します。セカンダリサブミラーは、最初のサブミラー以上のサイズでなければなりません。次のいずれかの方法を使用します。

注-2 番目のサブミラーとして使用するスライスには「root」というスライスタグを付ける必要があります。ルートスライスはスライス 0 である必要があります。スライスタグフィールドの構成方法については、[format\(1M\)](#) のマニュアルページを参照してください。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択し、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name number-of-stripes components-per-stripes component-names
```

注-オプションの説明については、手順 6 を参照してください。

- 8 次のいずれかの方法を使用して、1 面ミラーを作成します。

- Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、「アクション」、「ボリュームを作成」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name -m submirror-name
volume-name      作成するボリュームの名前を指定します。
-m              ミラーを作成することを指定します。
submirror-name   ミラーの最初のサブミラーになるコンポーネントの名前を指定
                 します。この例では、ルートスライスを含む RAID-0 ボ
                リュームです。
```

- 9 新しくミラー化したファイルシステムを再びマウントします。 `metaroot volume-name` コマンドを実行します (`volume-name` は、作成したミラーの名前で置き換えてください)。その後、システムをリブートします。

```
# metaroot volume-name
# reboot
```

詳細は、[metaroot\(1M\)](#) のマニュアルページを参照してください。

- 10 次の形式の `metattach` コマンドを使用して 2 番目のサブミラーを接続します。

```
# metattach volume-name submirror-name
volume-name      サブミラーを追加する RAID-1 ボリュームの名前を指定します。
submirror-name   ミラーに接続された 2 番目のサブミラーになるコンポーネントの
                 名前を指定します。
```

詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。

- 11 代替ブートパスを記録します。

プライマリサブミラーに障害が発生した場合にセカンダリサブミラーからブートするように、システムを構成する必要があります。セカンダリサブミラーを持つディスクからシステムがブートできるようにするには、そのディスクを代替ブートデバイスとして認識するようにシステムを構成します。

- a. 代替ブートデバイスのパスを調べます。ルート (`/`) ファイルシステムミラーに 2 番目のサブミラーとして接続されているスライスに対して `ls -l` コマンドを使用します。

```
# ls -l /dev/dsk/c1t1d0s0
lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/dsk/c1t1d0s0 -> ../
./devices/eisa/eha@1000,0/cmdk@1,0:a
```

- b. `/devices` ディレクトリに続く次の文字列 `/eisa/eha@1000,0/cmdk@1,0:a` を記録しておきます。これがデバイスツリーパスです。

注-システムが使用不可になる場合もあるため、この情報はシステム以外の場所
に書き留めるようにしてください。これによって、DCA を使ってシステムを
ブートする必要がある場合に、デバイスツリーパス情報の入力が簡単になりま
す。

c. eeprom コマンドを使用して、代替ブートパスを定義します。例:

```
# eeprom altbootpath=/eisa/eha@1000,0/cmdk@1,0:a
```

プライマリサブミラーに障害が発生した場合、システムはセカンダリサブミ
ラーからブートしようとします。自動的に2番目のディスクにフェイル
オーバーするように BIOS を構成できる場合、このブートプロセスは自動的に行
われます。そうでない場合は、BIOS に入って、セカンダリディスクからブートす
るよう構成する必要があります。ブートを開始したあと、システムはま
ず、bootpath デバイスからブートしようとします。ルートミラーのプライマリ
ブートディスクは動作していないため、システムは次に、altbootpath デバイス
からブートしようとします。BIOS の設定を構成する方法については、BIOS の
ユーザーガイドを参照してください。

システムが自動的にブートしない場合は、DCA を使用してセカンダリサブミ
ラーの選択を試みることができます。システムによっては、ブートプロセス中に
DCA に入ることもできます。このオプションが使用できない場合は、x86 ブート
フロッピーディスクからブートし、DCA を使用してセカンダリサブミラーを選択
する必要があります。オペレーティングシステムがブートしたあと、eeprom
bootpath の値を、代替ブートパスとして設定した値 (altbootpath の値) に変更し
ます。こうすれば、システムが自動的にブートします。

eeprom コマンドの使用の詳細は、[eeprom\(1M\)](#) のマニュアルページを参照して
ください。

例 11-7 x86: DCA を使用してルート (/) ファイルシステムからミラーを作成する

```
# metainit -f d1 1 1 c0t0d0s0
d1: Concat/Stripe is setup
# metainit d2 1 1 c0t1d0s0
d2: Concat/Stripe is setup
# metainit d0 -m d1
d0: Mirror is setup
# metaroot d0
# lockfs -fa
# reboot
...
# metattach d0 d2
d0: Submirror d2 is attached
# ls -l /dev/dsk/c0t1d0s0
lrwxrwxrwx 1 root root          88 Feb  8 15:51 /dev/dsk/c1t3d0s0 ->
../../../../devices/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw
# eeprom altbootpath=/pci@1f,0/pci@1,1/ide@3/dad@0,0:a,raw
# fdisk -b /usr/lib/fs/ufs/mboot /dev/dsk/c0t1d0p0
```

```
Total disk size is 31035 cylinders
Cylinder size is 1146 (512 byte) blocks
```

Partition	Status	Type	Cylinders		Length	%
			Start	End		
1	Active	Solaris	1	31034	31034	100

SELECT ONE OF THE FOLLOWING:

1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Enter Selection: 5

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot \
/usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdisk/c0t1d0s2
```

ルート (/) ファイルシステムをミラー化した場合のブート時の警告について

ルート (/) ファイルシステムをミラー化すると、エラーメッセージがコンソールに表示され、`/etc/syslog.conf` ファイルで定義にされたシステムログに記録されます。これらのエラーメッセージは、問題を示すものではありません。これらのエラーメッセージは、現在使用していないデバイスタイプごとに表示されます。未使用のモジュールを強制的にロードすることはできないからです。次のようなエラーメッセージが表示されます。

```
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forcload of
misc/md trans failed
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forcload of
misc/md raid failed
Jul 13 10:17:42 ifr genunix: [ID 370176 kern.warning] WARNING: forcload of
misc/md_hotspares failed
```

このようなエラーメッセージは無視してかまいません。

サブミラーに関する作業

▼ サブミラーを接続する方法

注- 「ラベル付きサブミラーをラベルなしミラーに接続することはできません」というエラーメッセージは、ミラーに RAID-0 ボリュームを接続できなかったことを意味します。ラベル付きボリューム (サブミラー) とは、その最初のコンポーネントがシリンダ 0 から始まるボリュームであり、一方、ラベルなしボリュームの最初のコンポーネントはシリンダ 1 から始まります。Solaris Volume Manager では、ラベル付きサブミラーのラベルが壊れるのを防ぐため、ラベル付きサブミラーのラベルなしミラーへの接続を許可しません。

始める前に [104 ページの「RAID-1 ボリュームの作成と保守」](#) を読んでください。

- 1 サブミラーとして使用するコンポーネント (連結またはストライプ) を特定します。このコンポーネントは、ミラー内の既存のサブミラー以上のサイズでなければなりません。サブミラーとなるボリュームをまだ作成していない場合は、[90 ページの「RAID-0 \(ストライプ\) ボリュームの作成」](#) または [92 ページの「RAID-0 \(連結\) ボリュームの作成」](#) を参照してください。
- 2 **root** 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 3 **metastat** コマンドを使用して、作業するミラーのステータスが「正常」状態であることを確認します。

```
# metastat mirror
```
- 4 次のいずれかの方法を使用して、サブミラーを接続します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - **metattach mirror submirror** コマンドを使用します。

```
# metattach mirror submirror
```

詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。
- 5 **metastat** コマンドを使用して、ミラーのステータスを表示します。

```
# metastat mirror
```


例 11-8 サブミラーを接続する

```
# metastat d30
d30: mirror
    Submirror 0: d60
    State: Okay
...
# metattach d30 d70
d30: submirror d70 is attached
# metastat d30
d30: mirror
    Submirror 0: d60
    State: Okay
    Submirror 1: d70
    State: Resyncing
    Resync in progress: 41 % done
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 2006130 blocks
...
```

この例では、サブミラー **d70** を 1 面ミラー **d30** に接続します。ミラーにサブミラーを接続したときに、2 面ミラーを作成します。ミラー **d30** は、最初、サブミラー **d60** から構成されています。サブミラー **d70** は RAID-0 ボリュームです。metastat コマンドでミラーのステータスが「正常」であることを確認してから、サブミラーを接続します。metattach コマンドを実行すると、新しいサブミラーと既存のミラーの再同期が実行されます。ミラーに追加のサブミラーを接続すると、メッセージが表示されます。ミラーの再同期が実行されていることを確認するには、metastat コマンドを使用します。

▼ サブミラーを切り離す方法

始める前に [104 ページの「RAID-1 ボリュームの作成と保守」](#) を読んでください。

- 1 **root** 権限を持っていることを確認します。すべてのデータについて、最新のバックアップがあることを確認します。
- 2 **metastat** コマンドを使用して、作業するミラーのステータスが「正常」状態であることを確認します。
- 3 次のいずれかの方法を使用して、サブミラーを切り離します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - metainit コマンドを使用してミラーからサブミラーを切り離します。

```
# metadetach mirror submirror
```

詳細は、[metadetach\(1M\)](#)のマニュアルページを参照してください。

例 11-9 サブミラーを切り離す

```
# metastat
d5: mirror
    Submirror 0: d50
    ...
# metadetach d5 d50
d5: submirror d50 is detached
```

この例では、ミラー `d5` にサブミラー `d50` があります。metadetach コマンドを使用してサブミラーを切り離します。d50 の基になるスライスは他の場所で再使用できます。ミラーからサブミラーを切り離すと、確認メッセージが表示されます。

▼ サブミラーをオフラインおよびオンラインにする方法

metaonline コマンドを実行できるのは、サブミラーが metaoffline コマンドによってオフラインにされている場合に限られます。metaonline コマンドを実行すると、Solaris Volume Manager によってサブミラーとミラーの再同期が自動的に開始されます。

注 - metaoffline コマンドの機能は、metadetach コマンドによって提供される機能と同様です。ただし、metaoffline コマンドでは、サブミラーとミラーの論理的な関連付けは切り離されません。

始める前に [104 ページの「RAID-1 ボリュームの作成と保守」](#) を読んでください。

- 1 root 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 2 次のいずれかの方法を使用して、サブミラーをオンラインまたはオフラインにします。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - metaoffline コマンドを使用してサブミラーをオフラインにします。

```
# metaoffline mirror submirror
```

詳細は、[metaoffline\(1M\)](#)のマニュアルページを参照してください。

- metaonline コマンドを使用してサブミラーをオンラインにします。

```
# metaonline mirror submirror
```

詳細は、[metaonline\(1M\)](#)のマニュアルページを参照してください。

例 11-10 サブミラーをオフラインにする

```
# metaoffline d10 d11
d10: submirror d11 is offlined
```

この例では、サブミラー `d11` をミラー `d10` からオフラインにします。読み取りは、もう一方のサブミラーから引き続き行われます。最初の書き込みが行われた時点でミラーは同期していない状態になります。この不整合の状態は、オフラインにしたサブミラーをオンラインに戻すと修正されます。

例 11-11 サブミラーをオンラインにする

```
# metaonline d10 d11d10: submirror d11 is onlined
```

この例では、サブミラー `d11` をミラー `d10` でオンラインに戻します。

▼ サブミラーの内のスライスを有効にする方法

始める前に 248 ページの「RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要」および 104 ページの「RAID-1 ボリュームの作成と保守」を読んでください。

- 1 root 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 2 次のいずれかの方法を使用して、サブミラー内のスライスを有効にします。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - `metareplace` コマンドを使用して、サブミラー内の障害が発生したスライスを有効にします。

```
# metareplace -e mirror failed-slice
```

`metareplace` コマンドを実行すると、修復または置換されたスライスとミラーのほかの部分との再同期が自動的に開始されます。

詳細は、[metareplace\(1M\)](#) のマニュアルページを参照してください。

例 11-12 サブミラーの内のスライスを有効にする

```
# metareplace -e d11 c1t4d0s7
d11: device c1t4d0s7 is enabled
```

この例では、ミラー `d11` のサブミラーに含まれるスライス `c1t4d0s7` にソフトウェアがあります。 `-e` オプションを指定した `metareplace` コマンドで、障害が発生したスライスを有効にします。

物理ディスクに障害が発生した場合は、そのディスクをシステム上で利用可能なほかのディスク (およびそのスライス) で置き換えることができます (144 ページの「サブミラー内のスライスを置換する方法」を参照)。あるいは、ディスクを修復または交換し、フォーマットした上で、この例のように、`-e` オプションを指定した `metareplace` コマンドを実行することもできます。

RAID-1 ボリュームの保守

▼ ミラーとサブミラーのステータスを表示する方法

始める前に RAID-1 ボリュームとサブミラーに関連するステータス情報の概要については、[107 ページの「保守作業を判断するためのサブミラーのステータスについて」](#)を参照してください。

- 次のいずれかの方法を使用して、ミラーやサブミラーのステータスをチェックします。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - ミラーに `metastat` コマンドを実行して、各サブミラーのステータスを表示します。

```
# metastat mirror
```

ミラーのパス番号、読み取りオプション、または書き込みオプションを変更する方法については、[142 ページの「RAID-1 ボリュームオプションを変更する方法」](#)を参照してください。

デバイスステータスのチェックの詳細は、[metastat\(1M\)](#)のマニュアルページを参照してください。

例 11-13 RAID-1 ボリュームのステータスをチェックする

metastat コマンドの出力例を次に示します。ミラー名を指定しないで metastat コマンドを使用すると、すべてのミラーのあらゆるステータスが表示されます。

```
# metastatd70: Mirror
  Submirror 0: d71
    State: Okay
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 12593637 blocks

d71: Submirror of d70
  State: Okay
  Size: 12593637 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Reloc Hot Spare
    c1t3d0s3         0           No  Okay       Yes
  Stripe 1:
    Device          Start Block  Dbase State      Reloc Hot Spare
    c1t3d0s4         0           No  Okay       Yes
  Stripe 2:
    Device          Start Block  Dbase State      Reloc Hot Spare
    c1t3d0s5         0           No  Okay       Yes

d0: Mirror
  Submirror 0: d1
    State: Okay
  Submirror 1: d2
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 5600 blocks

d1: Submirror of d0
  State: Okay
  Size: 5600 blocks
  Stripe 0:
    Device          Start Block  Dbase State      Hot Spare
    c0t2d0s7         0           No  Okay

...
```

ミラー名引数を指定して metastat コマンドを使用すると、そのミラーに対応する出力が表示されます。

```
metastat d70
d70: Mirror
  Submirror 0: d71
    State: Okay
```

```

Pass: 1
Read option: roundrobin (default)
Write option: parallel (default)
Size: 12593637 blocks

d71: Submirror of d70
State: Okay
Size: 12593637 blocks
Stripe 0:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t3d0s3        0           No  Okay        Yes
Stripe 1:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t3d0s4        0           No  Okay        Yes
Stripe 2:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t3d0s5        0           No  Okay        Yes

```

metastat コマンドは、ミラーのサブミラーごとに、そのステータス、「起動」行(エラーがある場合)、割り当てられたホットスペアプール(ホットスペアがある場合)、ブロック数、サブミラーの各スライスの情報を表示します。

▼ RAID-1 ボリュームオプションを変更する方法

始める前に [106 ページの「RAID-1 ボリュームオプションについて」](#)を確認してください。

- 1 root 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 2 次のいずれかの方法を使用して、RAID-1 オプションを変更します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - metaparam コマンドを使用して、ミラーのオプションを表示および変更します。

```
# metaparam [mirror options] mirror
```

ミラーオプションについては、[106 ページの「RAID-1 ボリュームオプションについて」](#)を参照してください。また、[metaparam\(1M\)](#) のマニュアルページも参照してください。

例 11-14 RAID-1 ボリュームの読み取りポリシーを変更する

```

# metaparam -r geometric d30
# metaparam d30
d30: mirror current parameters are:
    Pass: 1

```

```
Read option: geometric (-g)
Write option: parallel (default)
```

この例では、`-r` オプションでミラーの読み取りポリシーを `geometric` に変更します。

例 11-15 RAID-1 ボリュームの書き込みポリシーを変更する

```
# metaparam -w serial d40
# metaparam d40
d40: mirror current parameters are:
    Pass: 1
    Read option: roundrobin (default)
    Write option: serial (-S)
```

この例では、`-r` オプションでミラーの書き込みポリシーを `serial` に変更します。

例 11-16 RAID-1 ボリュームのパス番号を変更する

```
# metaparam -p 5 d50
# metaparam d50
d50: mirror current parameters are:
    Pass: 5
    Read option: roundrobin (default)
    Write option: parallel (default)
```

この例では、`-p` オプションでミラーのパス番号を `5` に変更します。

▼ RAID-1 ボリュームを拡張する方法

始める前に [104 ページの「RAID-1 ボリュームの作成と保守」](#) を読んでください。

- 1 **root** 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 2 次のいずれかの方法を使用して、ミラーを拡張します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - `metattach` コマンドを使用して、各サブミラーに追加のスライスを接続します。

```
# metattach submirror slice
```

ミラー内のすべてのサブミラーを拡張する必要があります。詳細は、[metattach\(1M\)](#) のマニュアルページを参照してください。

- 3 **metattach** コマンドを使用して、サブミラーのサイズに基づいてミラーのサイズを再計算させます。

```
# metattach mirror
```

例 11-17 マウントされたファイルシステムを持つ 2 面ミラーを拡張する

```
# metastat
d8: Mirror
  Submirror 0: d9
    State: Okay
  Submirror 1: d10
    State: Okay
...
# metattach d9 c0t2d0s5
d9: component is attached
# metattach d10 c0t3d0s5
d10: component is attached
# metattach d8
```

この例では、2つのディスクドライブをミラーの2つのサブミラーに連結することによって、ミラー化されたマウント済みのファイルシステムを拡張する方法を示します。ミラーは **d8** という名前で、2つのサブミラー **d9** と **d10** を含んでいます。

参照 UFS の場合は、ミラーボリュームに対して **growfs(1M)** コマンドを実行します。247 ページの「ファイルシステムを拡張する方法」を参照してください。

データベースなど、**raw** ボリュームを使用するアプリケーションは、追加されたストレージを独自の方法で拡張できなければなりません。

RAID-1 ボリュームのコンポーネント障害に対する処置

▼ サブミラー内のスライスを置換する方法

始める前に 248 ページの「RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要」および 104 ページの「RAID-1 ボリュームの作成と保守」を読んでください。

- 1 **root** 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 2 **metastat** コマンドを使用して、RAID-1 ボリュームとそのサブミラーのステータスを表示します。

```
# metastat mirror-name
```


- 3 次のいずれかの方法を使用して、サブミラー内のスライスを置換します。
- Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metareplace` コマンドを使用して、サブミラー内のスライスを置換します。

```
# metareplace mirror-name component-name
```

- `mirror-name` は、作成するボリュームの名前です。
- `component-name` は、置換するコンポーネントの名前です。

`mirror-name` 作成するボリュームの名前を指定します

`component-name` 置換するコンポーネントの名前を指定します

詳細は、次の例と `metainit(1m)` のマニュアルページを参照してください。

例 11-18 ミラー内の障害が発生したスライスを置換する

次の例では、障害が発生したスライスを置換する方法を示します。ただし、システムは、ホットスワップを使って障害が発生したディスクを自動的に置換するようには構成されていないものとします。ホットスワップの使用に関する詳細は、第 16 章「ホットスワップ(概要)」を参照してください。

```
# metastat d6
d6: Mirror
   Submirror 0: d16
     State: Okay
   Submirror 1: d26
     State: Needs maintenance
...
d26: Submirror of d6
     State: Needs maintenance
     Invoke: metareplace d6 c0t2d0s2 <new device>
...
# metareplace d6 c0t2d0s2 c0t2d2s2
d6: device c0t2d0s2 is replaced with c0t2d2s2
```

`metastat` コマンドを使用して、ミラー `d6` にサブミラー `d26` があり、そのスライスの状態が「保守が必要」であることを確認します。`metareplace` コマンドにより、`metastat` コマンドの出力中の「起動」行に従って、このスライスをシステム内の別の使用可能なスライスで置き換えます。スライスが置き換えられたことを示すメッセージが表示され、サブミラーの再同期が開始されます。

▼ サブミラーを置換する方法

始める前に 248 ページの「RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要」および 104 ページの「RAID-1 ボリュームの作成と保守」を読んでください。

- 1 root 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。

- 2 **metastat** コマンドを使用して、RAID-1 ボリュームとそのサブミラーのステータスを表示します。

```
# metastat mirror-name
```

- 3 次のいずれかの方法を使用して、サブミラーを置換します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。ミラーを選択します。次に、「アクション」、「プロパティ」の順に選択し、「サブミラー」タブをクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。

- **metadetach**、**metaclear**、**metatinit**、および **metattach** コマンドを使用して、サブミラー全体を置換します。

- a. **metadetach** コマンドを使用して、障害が発生したサブミラーをミラーから切り離します。

```
# metadetach -f mirror-name submirror
```

-f 切り離しを強制的に実行します

mirror-name ミラーの名前を指定します

submirror 切り離すサブミラーを指定します

- b. **metaclear** コマンドを使用して、サブミラーを削除します。

```
# metaclear -f submirror
```

-f サブミラーの削除を強制的に実行します

submirror 削除するサブミラーを指定します

- c. **metainit** コマンドを使用して、新しいサブミラーを作成します。

```
# metainit volume-name number-of-stripes components-per-stripe component-name
```

volume-name 作成するボリュームの名前を指定します。ボリュームの命名については、47 ページの「ボリューム名」を参照してください。

number-of-stripes 作成するストライプの数を指定します。

components-per-stripe 各ストライプに含めるコンポーネントの数を指定します。

component-names 使用されるコンポーネントの名前を指定します。この例ではルートスライス `c0t0d0s0` を使用します。

d. `metattach` コマンドを使用して、新しいサブミラーを接続します。

```
# metattach mirror submirror
```

例 11-19 ミラー内のサブミラーを置換する

次の例では、アクティブなミラー内のサブミラーを置換する方法を示します。

```
# metastat d20
d20: Mirror
    Submirror 0: d21
        State: Okay
    Submirror 1: d22
        State: Needs maintenance
...
# metadetach -f d20 d22
d20: submirror d22 is detached
# metaclear -f d22
d22: Concat/Stripe is cleared
# metainit d22 2 1 c1t0d0s2 1 c1t0d1s2
d22: Concat/Stripe is setup
# metattach d20 d22
d20: components are attached
```

この例では、`metastat` コマンドを使って、2面ミラー `d20` にサブミラー `d22` があり、その状態が「保守が必要」であることを確認します。この場合、サブミラー全体を削除し、作成し直す必要があります。`metadetach` コマンドに `-f` オプションを指定して、障害が発生したサブミラーを強制的にミラーから切り離します。`metaclear` コマンドは、サブミラーを削除します。`metainit` コマンドは、新しいスライスからサブミラー `d22` を再作成します。最後に、`metattach` コマンドは、作成し直したサブミラーを接続します。ミラーの再同期が自動的に開始されます。

新しいボリューム `d22` の構成は、置き換えるコンポーネントによって異なります。この場合は連結で十分に連結を置き換えることができます。しかし、ストライプの場合はパフォーマンスに影響する可能性があるため、連結が最適な置換とはなりません。

ミラーが1面ミラーとなっている間は、データの冗長性が一時的に失われます。

RAID-1 ボリュームの削除(ミラー化の解除)

▼ ファイルシステムのミラー化を解除する方法

システムの稼働中にアンマウントできるファイルシステムのミラー化を解除するには、次の手順を使用します。ルート (/)、/var、/usr、swap、またはシステムの稼働中はアンマウントできないその他のファイルシステムのミラー化を解除する場合は、[150 ページの「アンマウントできないファイルシステムのミラー化を解除する方法」](#)を参照してください。

始める前に [104 ページの「RAID-1 ボリュームの作成と保守」](#) を読んでください。

- 1 root 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。
- 2 少なくとも1つのサブミラーが「正常」状態であることを確認します。
`metastat mirror`
- 3 ファイルシステムをマウント解除します。
`umount /file-system`
- 4 サブミラーを切り離します(これは引き続きファイルシステム用に使用されます)。
`metadetach mirror submirror`
詳細は、[metadetach\(1M\)](#) のマニュアルページを参照してください。
- 5 ミラーと残りのサブコンポーネントを削除します。
`metaclear -r mirror`
詳細は、[metaclear\(1M\)](#) のマニュアルページを参照してください。
- 6 必要であれば、[手順4](#)で切り離したコンポーネントを使用するように、`/etc/vfstab` ファイルを編集します。
- 7 ファイルシステムを再マウントします。
`mount /file-system`

例 11-20 /opt ファイルシステムのミラー化を解除する

```
# metastat d4
d4: Mirror
   Submirror 0: d2
   State: Okay
```

```

Submirror 1: d3
  State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 2100735 blocks (1.0 GB)

d2: Submirror of d4
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c0t0d0s0      0          No         Okay   Yes

d3: Submirror of d4
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c1t0d0s0      0          No         Okay   Yes

...
# umount /opt
# metadetach d4 d2
d4: submirror d2 is detached
# metaclear -r d4
d4: Mirror is cleared
d3: Concat/Stripe is cleared
    (Edit the /etc/vfstab file so that the entry for /opt is changed from d4 to the underlying slice or volume)
# mount /opt

```

この例の /opt ファイルシステムは、2面ミラー d4 から構成されています。このミラーのサブミラーは d2 と d3 です。サブミラーはスライス /dev/dsk/c0t0d0s0 と /dev/dsk/c1t0d0s0 から構成されています。metastat コマンドで、少なくとも1つのサブミラーが「正常」状態であることを確認します。(「正常」状態のサブミラーが存在しないミラーは、最初に修復する必要があります。)ファイルシステムをアンマウントします。その後、サブミラー d2 を切り離します。metaclear -r コマンドで、ミラーともう1つのサブミラー d3 を削除します。

次に、/etc/vfstab ファイルで、/opt のエントリが基になるスライスを参照するように変更します。

この例では、/etc/vfstab ファイルに /opt ファイルシステムのエントリが次のように指定されています。

```
/dev/md/dsk/d4 /dev/md/rdisk/d4 /opt ufs 2 yes -
```

そのエントリを次のように変更します。

```
/dev/md/dsk/d2 /dev/md/rdisk/d2 /opt ufs 2 yes -
```

サブミラー名を使用することによって、ファイルシステムをボリュームにマウントしたままにできます。最後に、`/opt` ファイルシステムを再びマウントします。

`/etc/vfstab` ファイル内で `d4` の代わりに `d2` を使用したことによって、このミラーのミラー化を解除しています。`d2` は1つのスライスから構成されているため、このデバイスでボリュームをサポートしたくなければ、ファイルシステムをスライス名 (`/dev/dsk/c0t0d0s0`) にマウントできます。

▼ アンマウントできないファイルシステムのミラー化を解除する方法

ルート (`/`)、`/usr`、`/opt`、`swap` など、通常のシステム稼働中にはアンマウントできないファイルシステムのミラー化を解除するには、次のタスクを使用します。

- 1 **root** 権限を持っていること、およびすべてのデータの最新のバックアップを取っていることを確認します。

- 2 少なくとも1つのサブミラーが「正常」状態であることを確認します。

```
# metastat mirror
```

- 3 サブミラーを切り離します(これは引き続きファイルシステム用に使用されます)。

```
# metadetach mirror submirror
```

詳細は、[metadetach\(1M\)](#) のマニュアルページを参照してください。

- 4 ミラー化を解除するファイルシステムに応じて、次のいずれかのコマンドを使用します。

- `/usr`、`/opt`、または `swap` ファイルシステムの場合は、`/etc/vfstab` ファイル内のファイルシステムエントリを変更して、Solaris Volume Manager 以外のデバイス(スライス)が使用されるようにします。

- ルート (`/`) ファイルシステムの場合のみ:`metaroot` コマンドを実行します。

```
# metaroot rootslice
```

詳細は、[metaroot\(1M\)](#) のマニュアルページを参照してください。

- 5 システムをリブートします。

```
# reboot
```

- 6 残りのミラーとサブミラーを削除します。

```
# metaclear -r mirror
```

詳細は、[metaclear\(1M\)](#) のマニュアルページを参照してください。

例 11-21 ルート (/) ファイルシステムのミラー化を解除する

```

# metastat d0
d0: Mirror
  Submirror 0: d10
    State: Okay
  Submirror 1: d20
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 2100735 blocks (1.0 GB)

d10: Submirror of d0
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c0t3d0s0      0         No         Okay   Yes

d20: Submirror of d0
  State: Okay
  Size: 2100735 blocks (1.0 GB)
  Stripe 0:
    Device      Start Block  Dbase      State Reloc Hot Spare
    c1t3d0s0      0         No         Okay   Yes

# metadetach d0 d20
d0: submirror d20 is detached
# metaroot /dev/dsk/c0t3d0s0
# reboot
...
# metaclear -r d0
d0: Mirror is cleared
d10: Concat/Stripe is cleared
# metaclear d20
d20: Concat/Stripe is cleared

```

この例のルート (/) ファイルシステムは、2面ミラー `d0` です。このミラーのサブミラーは `d10` と `d20` です。サブミラーはスライス `/dev/dsk/c0t3d0s0` と `/dev/dsk/c1t3d0s0` から構成されています。 `metastat` コマンドで、少なくとも1つのサブミラーが「正常」状態であることを確認します。(「正常」状態のサブミラーが存在しないミラーは、最初に修復する必要があります。)サブミラー `d20` を切り離して、`d0` を1面ミラーにします。

「ルートスライス」は、ルート (/) ファイルシステムが含まれているスライスです。システムのブートに使用する `rootslice` を指定して `metaroot` コマンドを実行します。このコマンドによって `/etc/system` ファイルと `/etc/vfstab` ファイルが編集されます。このコマンドは、ルート (/) ファイルシステムのミラー化を指定する情報を削除します。

システムのリブート後、`metaclear -r` コマンドで、ミラーともう 1 つのサブミラー `d10` を削除します。最後の `metaclear` コマンドは、サブミラー `d20` を削除します。

例 11-22 swap ファイルシステムのミラー化を解除する

```
# metastat d1
d1: Mirror
   Submirror 0: d11
   State: Okay
   Submirror 1: d21
   State: Okay
...
# metadetach d1 d21
d1: submirror d21 is detached
   (Edit the /etc/vfstab file to change the entry for swap from metadvice to slice name)
# reboot
...
# metaclear -r d1
d1: Mirror is cleared
d11: Concat/Stripe is cleared
# metaclear d21
d21: Concat/stripe is cleared
```

この例の swap ファイルシステムは、2 面ミラー `d1` から構成されています。このミラーのサブミラーは `d11` と `d21` です。サブミラーはスライス `/dev/dsk/c0t3d0s1` と `/dev/dsk/c1t3d0s1` から構成されています。`metastat` コマンドで、少なくとも 1 つのサブミラーが「正常」状態であることを確認します。(「正常」状態のサブミラーが存在しないミラーは、最初に修復する必要があります。)サブミラー `d21` を切り離して、`d1` を 1 面ミラーにします。次に、`/etc/vfstab` ファイルを編集して、`swap` のエントリがサブミラー `d21` のスライスを参照するように変更します。

この例では、`/etc/vfstab` ファイルに `swap` ファイルシステムのエントリが次のように指定されています。

```
/dev/md/dsk/d4 /dev/md/rdisk/d4 /opt ufs 2 yes -
```

```
/dev/md/dsk/d1 - - swap - no -
```

そのエントリを次のように変更します。

```
/dev/dsk/c0t3d0s1 - - swap - no -
```

システムのリブート後、`metaclear -r` コマンドで、ミラーともう 1 つのサブミラー `d11` を削除します。最後の `metaclear` コマンドは、サブミラー `d21` を削除します。

RAID-1 ボリューム上でのデータのバックアップ

Solaris Volume Manager は「バックアップ製品」を意図しているわけではありません。しかし、Solaris Volume Manager は次のいずれも引き起こすことなく、ミラー化されたデータをバックアップする手段を提供します。

- ミラーのアンマウント
- ミラー全体のオフライン化
- システムの停止
- ユーザーに対するデータアクセス拒否

Solaris Volume Manager は、最初にサブミラーの1つをオフラインにすることによって、ミラー化されたデータをバックアップします。バックアップの間、ミラー化は一時的に使用できなくなります。バックアップの完了と同時に、サブミラーがオンラインに戻り、再同期が実行されます。

注-UFSのスナップショット機能で、ファイルシステムをオフラインにすることなく、システムをバックアップすることもできます。サブミラーの切り離しや、あとでミラーを再同期させることによるパフォーマンス低下を伴わずに、バックアップを実行できます。UFSのスナップショット機能によるバックアップを実行する前に、UFSファイルシステム上の使用可能領域が十分かどうかを確認してください。詳細は、『Oracle Solarisの管理: デバイスとファイルシステム』の第20章「UFSスナップショットの使用(手順)」を参照してください。

▼ RAID-1 ボリュームのオンラインバックアップを実行する方法

この手順は、ルート(/)ファイルシステム以外のすべてのファイルシステムに使用できます。このタイプのバックアップは、アクティブなファイルシステムの「スナップショット」を作成することに注意してください。ファイルシステムへの書き込みをロックしたときのファイルシステムの使用状況によっては、バックアップしたファイルがディスク上の実際のファイルに対応しないことがあります。

この手順には、次の制限があります。

- この手順を2面ミラーに対して使用すると、1つのサブミラーをバックアップのためにオフラインにしたときに、データの冗長性が失われます。多面ミラーにはこの問題はありません。
- バックアップの完了後に、再接続されたサブミラーを再同期するときに、システムにある程度のオーバーヘッドが生じます。

この手順の概要は次のとおりです。

- ファイルシステムへの書き込みをロックします (UFS のみ)。ルート (/) はロックしないでください。
- キャッシュからすべてのデータをディスクにフラッシュします。
- `metadetach` コマンドを使用して、このミラーの1つのサブミラーを切り離します
- ファイルシステムのロックを解除します
- `fsck` コマンドを使用して、切り離れたサブミラー上のファイルシステムをチェックします
- 切り離れたサブミラー上のデータをバックアップします
- `metattach` コマンドを使用して、切り離れたサブミラーをミラーに再び接続します

注-このような手順を定常的に使用する場合は、これをスクリプトにしておくことと実行が容易になります。

ヒント-より安全な方法としては、ミラーに3番目または4番目のサブミラーを接続し、これを再同期し、バックアップに使用します。この手法では、データの冗長性が常に保たれます。

- 1 ミラーが「正常」状態であることを確認します。
ミラーが「保守」状態の場合は、まずそれを修復する必要があります。
- 2 キャッシュからデータと UFS ロギングデータをディスクにフラッシュし、このファイルシステムを書き込みをロックします。

```
# /usr/sbin/lockfs -w mount-point
```

書き込みをロックする必要があるのは UFS ボリュームだけです。このボリュームがデータベース管理ソフトウェアなどのアプリケーション用に raw デバイスとして設定されている場合は、`lockfs` コマンドを実行する必要はありません。ただし、ベンダー提供の適切なユーティリティを実行してバッファをフラッシュしたり、アクセスをロックする必要がある場合もあります。



注意-ルート (/) ファイルシステムを書き込みはロックしないでください。ルート (/) ファイルシステムを書き込みをロックすると、システムがハングします。ルート (/) ファイルシステムをバックアップしている場合は、この手順をスキップします。

- 3 ミラーから1つのサブミラーを切り離します。

```
# metadetach mirror submirror
```

mirror ミラーのボリューム名です。

submirror 切り離すサブミラー (ボリューム) のボリューム名です。

読み取りは、もう一方のサブミラーから引き続き行われます。最初の書き込みが行われた時点でミラーは同期していない状態になります。この不整合の状態は、切り離されたサブミラーが手順7で再び接続された時点で修復されます。

- 4 ファイルシステムのロックを解除し、書き込みを再開します。

```
# /usr/sbin/lockfs -u mount-point
```

手順2で使用したベンダー提供のユーティリティーを使用して、必要なロック解除手順を実行しなければならない場合があります。

- 5 **fsck** コマンドを使用して、切り離したサブミラー上のファイルシステムをチェックします。この手順により、クリーンなバックアップが保証されます。

```
# fsck /dev/md/rdisk/name
```

- 6 オフラインにしたサブミラーのバックアップを実行します。

ufsdump コマンド、または通常使用しているバックアップユーティリティーを使用します。**ufsdump** コマンドを使ってバックアップを実行する方法については、[320 ページ](#)の「**ufsdump** コマンドによるマウント済みファイルシステムのバックアップの実行」を参照してください。

注 - 適切なバックアップを確実に行うには、*raw* ボリューム名 (*/dev/md/rdisk/d4* など) を使用します。*raw* ボリューム名を使用すると、2G バイトを超えるストレージにアクセスできます。

- 7 サブミラーを接続します。

```
# metattach mirror submirror
```

Solaris Volume Manager によってサブミラーとミラーの再同期が自動的に開始されます。

例 11-23 RAID-1 ボリュームのオンラインバックアップを実行する

この例では、ミラー *d1* を使用します。このミラーはサブミラー *d2*、*d3*、および *d4* からなります。サブミラー *d3* を切り離して、そのバックアップを取ります。この間、サブミラー *d2* と *d4* はオンラインのままです。このミラーにあるファイルシステムは */home1* です。

```
# metastat d1
d1: Mirror
    Submirror 0: d2
        State: Okay
    Submirror 1: d3
```

```
State: Okay
Submirror 1: d4
State: Okay
...

# /usr/sbin/lockfs -w /home1
# metadetach d1 d3
# /usr/sbin/lockfs -u /home1
# /usr/sbin/fsck /dev/md/rdisk/d3
(Perform backup using /dev/md/rdisk/d3)
# metattach d1 d3
```

ソフーパーティション(概要)

この章では、Solaris Volume Manager のソフーパーティションについて説明します。関連タスクについては、第13章「ソフーパーティション(タスク)」を参照してください。

この章では、次の内容について説明します。

- 157 ページの「ソフーパーティションの概要」
- 158 ページの「ソフーパーティションの構成のガイドライン」

ソフーパーティションの概要

ディスクのストレージ容量が大きくなると、ディスクアレイによって Solaris システムに提供される論理デバイスが大きくなります。管理しやすいファイルシステムやパーティションのサイズを作成するため、ユーザーはディスクや論理ボリュームを 8 つ以上のパーティションに分割する必要がある場合があります。Solaris Volume Manager のソフーパーティション機能はこのニーズに対処します。

Solaris Volume Manager はディスクセットあたり最大 8192 の論理ボリュームをサポートできます。この数値には、ローカルまたは無指定のディスクセットも含まれます。Solaris Volume Manager は必要に応じて、動的にボリュームを構成します。

ソフーパーティションを使用して、ディスクスライスや論理ボリュームを必要な数だけのパーティションに分割できます。各区画または「ソフーパーティション」には、ストライプやミラーなど、他のストレージボリュームと同様に、名前を付ける必要があります。名前を付けたソフーパーティションには、これが別のボリュームに含まれていない限り、ファイルシステムなどのアプリケーションからアクセスできます。ボリュームに含まれたソフーパーティションには、直接アクセスできなくなります。

ソフーパーティションは、ディスクスライス上に直接配置することも、ミラー、ストライプ、RAID-5 ボリューム上に配置することもできます。ソフーパーティション

は他のボリュームの上と下の両方に配置することはできません。たとえば、ソフトパーティション上にミラーのあるストライプを構築し、その上にソフトパーティションを構築することはできません。

ソフトパーティションはファイルシステムなどのアプリケーションからは1つの連続した論理ボリュームに見えます。ただし、実際には、ソフトパーティションは、基盤のメディアの任意の場所にある一連の「エクステント」から構成されています。ソフトパーティションに加えて、システムに重大な障害が発生した場合に復元を促進するために、ディスク上のエクステントヘッダー（「システム復元データ領域」ともいう）に、ソフトパーティションに関する情報が記録されます。

ソフトパーティションの構成のガイドライン

- ソフトパーティションとして使用されているスライスを他の目的で使用することはできません。
- ディスクをパーティション分割し、その結果のスライスにファイルシステムを構築する場合、後で、ディスクフォーマットを変更するか破棄しないと、スライスを拡張することはできません。ソフトパーティションの動作は異なります。ソフトパーティションでは、他のソフトパーティション上のデータを移動したり、破棄したりすることなく、基となるデバイスの容量までソフトパーティションを拡張できます。
- 技術的にソフトパーティションのエクステントをディスク上の任意の場所に手動で配置できますが、システムに自動的にそれらを配置させるようにしてください。手動で配置されたエクステントの例については、[236 ページの「Solaris Volume Manager 構成の表示」](#)の `metastat` コマンドの出力を参照してください。
- ソフトパーティションは、任意のスライスに構築できます。ただし、ディスクレベルでソフトパーティションを使用するには、ディスク全体を占有する単一のスライスを作成し、そのスライスにソフトパーティションを作成するのが、もっとも効率的な方法です。
- ソフトパーティションの最大サイズはスライスのサイズや、それが構築されている論理ボリュームのサイズに制限されます。この制限のため、ディスクスライス上にボリュームを構築してから、そのボリューム上にソフトパーティションを構築してください。この方法により、ボリュームにコンポーネントを追加してから、必要に応じてソフトパーティションを拡張することができます。
- 最大の柔軟性と高可用性のため、ディスクスライスに RAID-1 (ミラー) または RAID-5 ボリュームを作成してから、そのミラーまたは RAID 5 ボリューム上にソフトパーティションを作成します。

シナリオ-ソフトパーティション

ソフトパーティションは、大きなストレージ領域を管理しやすい領域に分割するためのツールを備えています。たとえば、他のシナリオ(110ページの「シナリオ-RAID-1 ボリューム(ミラー)」または175ページの「シナリオ-RAID-5 ボリューム」)では、大容量ストレージアグリゲーションによって、数Gバイトの冗長ストレージを実現しました。しかし、可能性のあるシナリオの多くは、最初からそれほど大量の領域が必要になることはないと考えられます。ソフトパーティションにより、そのストレージ領域を管理しやすいパーティションに分割できます。それらのパーティションには、それぞれ完全なファイルシステムを格納できます。たとえば、RAID-1 または RAID-5 ボリューム上に 1000 のソフトパーティションを作成して、各ユーザーが個別のファイルシステムにホームディレクトリを持てるようにすることができます。ユーザーがより多くの領域を必要とする場合は、そのソフトパーティションを拡張するだけです。

◆◆◆ 第 13 章

ソフトパーティション(タスク)

この章では、Solaris Volume Manager のソフトパーティションに関連するタスクの実行について説明します。ソフトパーティションに関連する概念については、第 12 章「ソフトパーティション(概要)」を参照してください。

ソフトパーティション(タスクマップ)

次のタスクマップは、Solaris Volume Manager のソフトパーティションを管理するために必要な手順を示します。

タスク	説明	参照先
ソフトパーティションを作成する	Solaris Volume Manager の GUI または <code>metainit</code> コマンドを使用して、ソフトパーティションを作成します。	162 ページの「ソフトパーティションを作成する方法」
ソフトパーティションのステータスをチェックする	Solaris Volume Manager の GUI または <code>metastat</code> コマンドを使用して、ソフトパーティションのステータスをチェックします。	163 ページの「ソフトパーティションのステータスをチェックする方法」
ソフトパーティションを拡張する	Solaris Volume Manager の GUI または <code>metattach</code> コマンドを使用して、ソフトパーティションを拡張します。	164 ページの「ソフトパーティションを拡張する方法」
ソフトパーティションを削除する	Solaris Volume Manager の GUI または <code>metaclear</code> コマンドを使用して、ソフトパーティションを削除します。	165 ページの「ソフトパーティションを削除する方法」

ソフトパーティションの作成

▼ ソフトパーティションを作成する方法

始める前に [158 ページの「ソフトパーティションの構成のガイドライン」](#)を確認してください。

- 次のいずれかの方法を使用して、ソフトパーティションを作成します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「アクション」、「ボリュームを作成」の順に選択します。次に、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用して、ソフトパーティションを作成します。

```
# metainit [-s diskset] soft-partition -p [-e] component size
```

`-sdiskset` 使用するディスクセットを指定します。-s を指定しない場合は、ローカル(デフォルト)のディスクセットが使用されます。

`-p` ソフトパーティションを構成することを指定します。

`-e` ディスク全体を再フォーマットすることを指定します。ディスクをフォーマットすることによって、ディスクの大部分を占めるスライス 0 が得られます。ディスクをフォーマットすることによって、4M バイト以上のサイズのスライス 7 も得られます。スライス 7 には、状態データベースの複製が格納されます。

`soft-partition` ソフトパーティションの名前を指定します。名前の形式は `dnnn` で、`nnn` は 0 から 8192 の範囲内の数です。

`component` ソフトパーティションの作成に使用するディスク、スライス、または論理ボリュームを指定します。ソフトパーティションのヘッダーがコンポーネントの先頭部分に書き込まれるため、コンポーネントにあるデータはすべて破壊されます。

`size` ソフトパーティションのサイズを指定します。数値のあとに次のいずれかを付加してサイズを指定します。

- M または m (メガバイト)
- G または g (ギガバイト)
- T または t (テラバイト)
- B または b (ブロック数(セクター数))

詳細は、次の例と `metainit(1m)` のマニュアルページを参照してください。

例 13-1 ソフトパーティションを作成する

次の例では、d20 という名前の 4G バイトのソフトパーティションを c1t3d0s2 に作成します。

```
# metainit d20 -p c1t3d0s2 4g
```

例 13-2 ディスク全体をソフトパーティションに使用する

次の例では、ソフトパーティションを作成し、ディスク c1t2d0 をフォーマットします。この処理によって、そのディスク上のあらゆるデータが破壊され、スライス 0 に新しいソフトパーティションが作成されます。

```
# metainit d7 -p -e c1t2d0 1G
```

ソフトパーティションの保守

ソフトパーティションの保守は、他の論理ボリュームの保守と同じです。

▼ ソフトパーティションのステータスをチェックする方法

始める前に [158 ページの「ソフトパーティションの構成のガイドライン」](#) を読んでください。

- 次のいずれかの方法を使用して、ソフトパーティションのステータスをチェックします。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。モニターするソフトパーティションを選択します。次に、「アクション」、「プロパティ」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metastat` コマンドを使用して、既存の構成を表示します。

```
# metastat soft-partition
```

`soft-partition` チェックするパーティションの名前を指定します。

例 13-3 ソフトパーティションのステータスをチェックする

次の例では、ソフトパーティション d1 のステータスをチェックします。このソフトパーティションはエクステンタが 2 つあり、RAID-1 ボリューム d100 上に作成されています。

```
# metastat d1
d1: soft partition
   component: d100
   state: OKAY
   size: 42674285 blocks
      Extent          Start Block          Block Count
      0                10234                40674285
      1                89377263             2000000
d100: Mirror
     Submirror 0: d10
     State: OKAY
     Read option: roundrobin (default)
     Write option: parallel (default)
     Size: 426742857 blocks

d10: Submirror of d100
     State: OKAY
     Hot spare pool: hsp002
     Size: 426742857 blocks
     Stripe 0: (interlace: 32 blocks)
        Device          Start Block  Dbase State      Hot Spare
        c3t3d0s0        0           No    Okay
```

▼ ソフトパーティションを拡張する方法

ソフトパーティション上にほかの論理ボリュームが構築されていない場合は、そのソフトパーティションに領域を追加できます。空き領域を見つけ、パーティションの拡張に使用します。既存のデータは移動されません。

注-ソフトパーティションを使用して別のボリュームを作成している場合 (RAID-0 ボリュームのコンポーネントの場合など)、そのソフトパーティションは拡張できません。ソフトパーティションを収容するデバイスの領域を増やすことが目的であれば、通常、収容デバイスに他のボリュームを連結することによって目的を達成できます。詳細は、[93 ページの「ストレージ容量の拡張」](#)を参照してください。

始める前に [158 ページの「ソフトパーティションの構成のガイドライン」](#) を読んでください。

- 次のいずれかの方法を使用して、ソフトパーティションを拡張します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。拡張するソフトパーティションを選択し、「アクション」、「プロパティ」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metattach` コマンドを使用して、ソフトパーティションに領域を追加します。

```
# metattach [-s diskset] soft-partition size
```

<i>diskset</i>	ソフトパーティションが含まれているディスクセットの名前を指定します。
<i>soft-partition</i>	既存のソフトパーティションの名前を指定します。
<i>size</i>	追加するストレージ領域の量を指定します。

例 13-4 ソフトパーティションを拡張する

次の例では、ソフトパーティションに領域を追加する方法を示します。さらに、ソフトパーティションがオンライン状態であり、マウントされているときに、**growfs** コマンドを使ってファイルシステムを拡張します。

```
# mount /dev/md/dsk/d20 /home2
# metattach d20 10g
# growfs -M /home2 /dev/md/rdisk/d20
```

growfs コマンドの詳細については、246 ページの「growfs コマンドを使用したファイルシステムの拡張」を参照してください。

▼ ソフトパーティションを削除する方法

始める前に 158 ページの「ソフトパーティションの構成のガイドライン」を読んでください。

- 次のいずれかの方法を使用して、ソフトパーティションを削除します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。削除するソフトパーティションを選択します。次に、「アクション」、「プロパティ」の順に選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次のいずれかの形式の **metaclear** コマンドを使用して、ソフトパーティションを削除します。

```
# metaclear [-s diskset] component
# metaclear [-s diskset] -r soft-partition
# metaclear [-s diskset] -p component
```

<i>diskset</i>	ソフトパーティションが含まれているディスクセットを指定します。
<i>soft-partition</i>	削除するソフトパーティションを指定します。
-r	論理ボリュームを再帰的に削除することを指定します(ただし、他のボリュームに使用されているボリュームは除きます)。
-p	指定したコンポーネント上のすべてのソフトパーティションを削除することを指定します(ただし、開かれているソフトパーティションは除きます)。

component すべてのソフトパーティションを削除するコンポーネントを指定します。

例 13-5 ソフトパーティションを削除する

この例では、`c1t4d2s0` 上のすべてのソフトパーティションを削除する方法を示します。

```
# metaclear -p c1t4d2s0
```

RAID-5 ボリューム (概要)

この章では、Solaris Volume Manager の RAID-5 ボリュームに関する概念について説明します。関連タスクの実行の詳細については、第 15 章「RAID-5 ボリューム (タスク)」を参照してください。

この章では、次の内容について説明します。

- 167 ページの「RAID-5 ボリュームの概要」
- 171 ページの「RAID-5 ボリュームを作成するための背景情報」
- 172 ページの「RAID-5 ボリュームのステータスのチェックの概要」
- 174 ページの「RAID-5 ボリューム内のスライスの交換と有効化の概要」
- 175 ページの「シナリオ - RAID-5 ボリューム」

RAID-5 ボリュームの概要

RAID レベル 5 はストライプ化と似ていますが、パリティデータがすべてのコンポーネント (ディスクまたは論理ボリューム) に分散されます。コンポーネントで障害が発生した場合、障害の発生したコンポーネントのデータは、他のコンポーネントに分散されたデータおよびパリティ情報から再構築できます。Solaris Volume Manager で、RAID-5 ボリュームは RAID レベル 5 をサポートするボリュームです。

RAID-5 ボリュームでは、冗長情報 (パリティ) を格納するために、ボリューム内の 1 つのコンポーネントに相当するストレージ容量を使用します。このパリティ情報には、残りの RAID-5 ボリュームのコンポーネントに格納されているユーザーデータに関する情報が含まれます。つまり、3 つのコンポーネントがある場合、1 つのコンポーネント相当がパリティ情報に使われます。5 つのコンポーネントがある場合、1 つのコンポーネント相当がパリティ情報に使われます。パリティ情報は、ボリューム内のすべてのコンポーネントに分散されます。ミラーと同様に、RAID-5 ボリュームでは、データの可用性が向上しますが、ハードウェアの点でコストは最小で、書き込み操作に与える影響もわずかです。ただし、RAID-5 ボリュームはルート (/)、/usr、および swap ファイルシステム、または他の既存のファイルシステムに使用できません。

既存のコンポーネントを置換すると、Solaris Volume Manager によって自動的に RAID-5 ボリュームが置換されます。さらに、システムの障害やパニックが発生した場合のリブート時にも Solaris Volume Manager によって RAID-5 ボリュームが再同期されます。

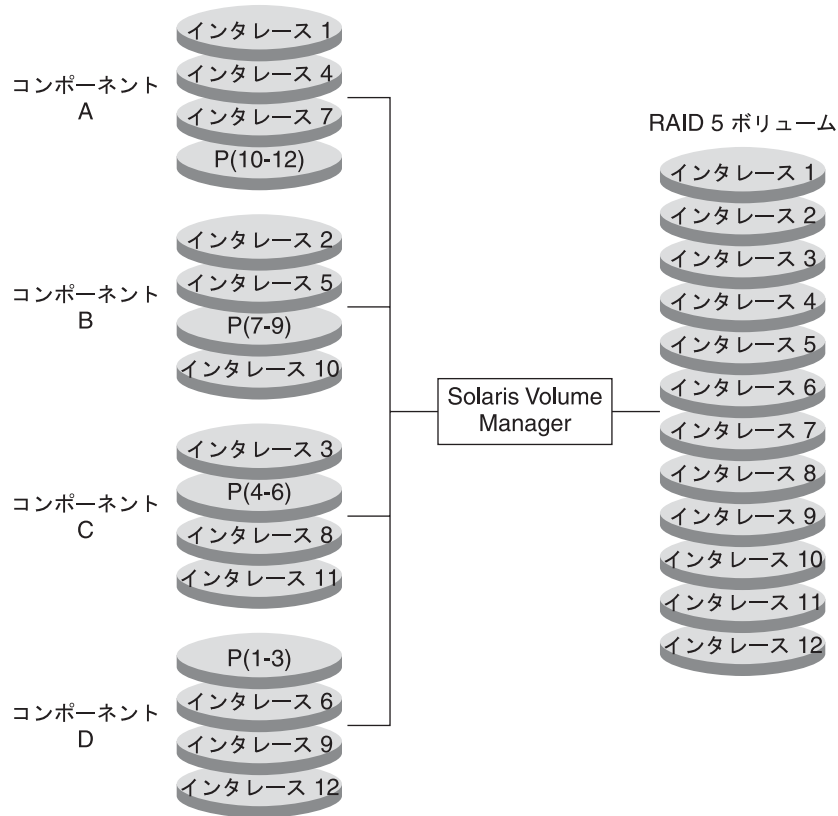
例 — RAID-5 ボリューム

図 14-1 に、4つのディスク (コンポーネント) から構成される RAID-5 ボリュームを示します。

最初の3つのデータセグメントは、コンポーネント A (インタレース 1)、コンポーネント B (インタレース 2)、コンポーネント C (インタレース 3) に書き込まれます。次に書き込まれるデータセグメントはパリティセグメントです。このパリティセグメントはコンポーネント D (P 1-3) に書き込まれます。このセグメントは、最初の3つのデータのセグメントの排他的 OR で構成されます。次の3つのデータセグメントは、コンポーネント A (インタレース 4)、コンポーネント B (インタレース 5)、コンポーネント D (インタレース 6) に書き込まれます。さらに、別のパリティセグメントがコンポーネント C (P 4-6) に書き込まれます。

このデータの書き込みとパリティセグメントのパターンにより、データとパリティの両方が、RAID-5 ボリュームのすべてのディスクに分散されます。各ドライブは個別に読み取ることができます。パリティは、単一のディスクの障害に対して保護します。この例の各ディスクが 2G バイトである場合、RAID-5 ボリュームの総容量は 6G バイトになります。1 ドライブ相当の領域がパリティに割り当てられます。

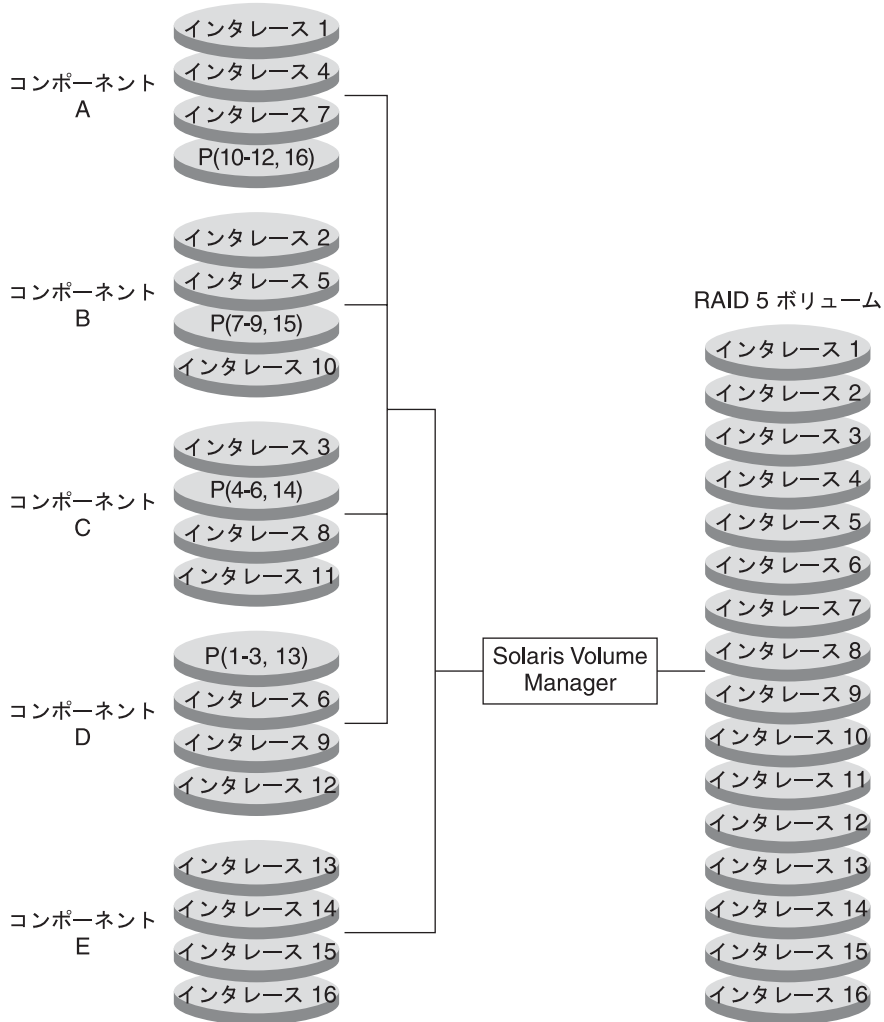
図 14-1 RAID-5 ボリュームの例



例 — 連結 (拡張) RAID-5 ボリューム

次の図に、最初に4つのディスク (コンポーネント) から構成された RAID-5 ボリュームの例を示します。5番目のディスクがボリュームに動的に連結され、RAID-5 ボリュームが拡張されています。

図 14-2 拡張された RAID-5 ボリュームの例



最初の RAID-5 ボリュームが作成されるときにパリティ領域が割り当てられます。実際のパリティブロックは、I/O を分散するために、元のすべてのコンポーネントに分散されますが、1つのコンポーネント相当の領域がパリティに割り当てられます。追加のコンポーネントが RAID-5 ボリュームに連結されると、追加の領域はすべてデータに割り当てられます。新しいパリティブロックは割り当てられません。ただし、連結されたコンポーネント上のデータは、パリティ計算に含まれるため、データは単一デバイスの障害に対して保護されます。

連結した RAID-5 ボリュームは長期間の使用には適しません。連結した RAID-5 ボリュームを使用するのは、これよりも大きな RAID-5 ボリュームを再構成できるようになるまでです。次に、大容量ボリュームにデータをコピーします。

注-RAID-5 ボリュームに新しいコンポーネントを追加すると、Solaris Volume Manager は、そのコンポーネントのすべてのブロックを「ゼロ」にします。このプロセスにより、パリティが新しいデータを保護します。データが追加の領域に書き込まれると、Solaris Volume Manager はそのデータをパリティ計算に含めます。

RAID-5 ボリュームを作成するための背景情報

RAID-5 ボリュームを操作するときは、171 ページの「RAID-5 ボリュームの要件」と172 ページの「RAID-5 ボリュームのガイドライン」を考慮してください。RAID-5 ボリュームの構成には、多くのストライプ化のガイドラインも適用されます。87 ページの「RAID-0 ボリュームの要件」を参照してください。

RAID-5 ボリュームの要件

- RAID-5 ボリュームは、少なくとも3つのコンポーネントから構成されている必要があります。ただし、RAID-5 ボリュームに含まれるコンポーネントの数が多くなるほど、コンポーネントに障害が発生したときに、読み取りと書き込みの操作に長くかかります。
- RAID-5 ボリュームをストライプ化、連結、ミラー化することはできません。
- 既存のファイルシステムが格納されているコンポーネントから RAID 5 ボリュームを作成しないでください。そうすると、RAID-5 の初期化プロセス時にデータが消去されます。
- RAID-5 ボリュームを作成する際に、インタレース値を定義できます。指定しない場合、インタレース値のデフォルトは16Kバイトに設定されます。この値はほとんどのアプリケーションに適切です。
- RAID-5 ボリューム(ホットスベアなし)は、1つのコンポーネントの障害しか処理できません。
- RAID-5 ボリュームを作成するときは、別個のコントローラにあるコンポーネントを使用します。コントローラと、関連付けられているケーブルの方がディスクよりも頻繁に故障する傾向があります。
- 同じサイズのコンポーネントを使用します。サイズが異なるコンポーネントで RAID-5 ボリュームを作成すると、使われないディスク領域が発生します。

RAID-5 ボリュームのガイドライン

- パリティ計算の複雑さのため、書き込みが約 20 パーセントを超えるボリュームは、RAID-5 ボリュームにすべきではないと考えられます。書き込みが頻繁なボリュームでデータの冗長性が必要な場合は、ミラー化を検討してください。
- RAID-5 ボリューム内のさまざまなコンポーネントがさまざまなコントローラ上にあり、ボリュームへのアクセスが主に大容量の順次アクセスである場合は、インタレース値を 32 K バイトに設定すると、パフォーマンスが向上することがあります。
- RAID-5 ボリュームに追加のコンポーネントを連結することによってボリュームを拡張できます。既存の RAID-5 ボリュームに新しいコンポーネントを連結すると、連結上のデータが順次処理されるため、ボリュームの全体的なパフォーマンスが低下します。データは、すべてのコンポーネントにストライプ化されるわけではありません。ボリュームの元のコンポーネントでは、すべてのコンポーネントにデータとパリティがストライプ化されます。このストライプ化は連結されたコンポーネントには行われません。ただし、コンポーネント I/O 時はパリティが使用されるため、エラーが発生してもデータは復元されます。結果の RAID-5 ボリュームは引き続き 1 つのコンポーネントの障害を処理します。

連結されたコンポーネントは、どの領域にもパリティがストライプ化されないという点でも異なります。そのため、コンポーネントの全内容をデータに使用できません。

コンポーネントを連結すると、大容量の書き込みや順次書き込みのパフォーマンスの向上は失われます。

- データブロックをゼロで初期化しなくても、RAID-5 ボリュームを作成することができます。そのためには、次のいずれかの操作を行ないます。
 - `metainit` コマンドと `-k` オプションを使います。`-k` オプションを指定すると、RAID 5 ボリュームが初期化なしで再作成され、ディスクブロックが「正常」状態に設定されます。このオプションは、危険な可能性があり、ボリューム内のディスクブロックにエラーがあると、データの改ざんの可能性など、Solaris Volume Manager が予期せぬ動作を起こします。
 - デバイスを初期化し、テープからデータを復元します。詳細については、`metainit(1m)` のマニュアルページを参照してください。

RAID-5 ボリュームのステータスのチェックの概要

RAID-5 ボリュームのステータスをチェックするには、ボリュームの状態とボリュームのスライスの状態を調べます。スライスの状態は、RAID-5 ボリュームのエラーのトラブルシューティングを行なう場合に、もっとも具体的な情報を提供します。RAID-5 ボリュームの状態は、「正常」や「保守」などの一般的なステータス情報しか提供しません。

RAID-5 ボリュームの状態で、「保守」状態が報告された場合は、スライスの状態を参照してください。スライスの状態は、スライスが「保守」状態か「最後にエラー」状態かを具体的に報告します。スライスが「保守」状態にあるか、「最後にエラー」状態にあるかによって、異なる復元作業を実行します。「保守」状態のスライスしかない場合は、データの損失なく修復できます。「保守」状態のスライスと「最後にエラー」状態のスライスがある場合は、データが破損している可能性があります。まず「保守」状態のスライスを修正してから、「最後にエラー」状態のスライスを修正する必要があります。

次の表に、RAID-5 ボリュームの状態を説明します。

表 14-1 RAID-5 ボリュームの状態

状態	意味
初期化中	スライスはすべてのディスクブロックをゼロにしています。このプロセスは、RAID 5 ボリュームの性質上、データとパリティのインタレーストライブ化に関して必要です。 状態が「正常」に変わったら、初期化プロセスが完了しており、デバイスを開くことができます。それまで、アプリケーションはエラーメッセージを受け取ります。
正常	デバイスは使用する準備が整っており、現在エラーはありません。
保守	I/O エラーかオープンエラーのため、スライスが障害としてマークされています。これらのエラーは、読み取り/書き込み処理中に発生しました。

次の表に、RAID-5 ボリュームのスライスの状態と、可能な対策を説明します。

表 14-2 RAID-5 スライスの状態

状態	意味	アクション
初期化中	スライスはすべてのディスクブロックをゼロにしています。このプロセスは、RAID 5 ボリュームの性質上、データとパリティのインタレーストライブ化に関して必要です。	通常、ありません。このプロセス中に I/O エラーが発生すると、デバイスは「保守」状態になります。初期化に失敗すると、このボリュームは「初期化失敗」状態になり、スライスは「保守」状態になります。この場合は、ボリュームをクリアしてから再作成してください。
正常	デバイスは使用する準備が整っており、現在エラーはありません。	なし。必要に応じて、スライスを追加または交換できます。

表 14-2 RAID-5 スライスの状態 (続き)

状態	意味	アクション
再同期中	スライスは再同期中です。エラーが発生しましたが、修正されており、スライスが有効になっているか、スライスが追加されています。	必要であれば、再同期が終了するまで RAID 5 ボリュームのステータスをモニターします。
保守	I/O エラーかオープンエラーのため、1つのスライスが障害としてマークされています。これらのエラーは、読み取り/書き込み処理中に発生しました。	障害の発生したスライスを有効にするか、交換します。182 ページの「RAID-5 ボリューム内のコンポーネントを有効にする方法」または183 ページの「RAID-5 ボリューム内のコンポーネントを置き換える方法」を参照してください。metastat コマンドは、invoke 復元メッセージと metareplace コマンドによって実行する適切な対策を表示します。
保守/最後にエラー	複数のスライスでエラーが発生しました。障害が発生したスライスの状態は「保守」か「最後にエラー」です。この状態では、「保守」状態のスライスには I/O が試みられません。ただし、「最後にエラー」とマークされているスライスには I/O が試みられ、その結果が I/O リクエストの全体的なステータスになります。	障害の発生したスライスを有効にするか、交換します。182 ページの「RAID-5 ボリューム内のコンポーネントを有効にする方法」または183 ページの「RAID-5 ボリューム内のコンポーネントを置き換える方法」を参照してください。metastat コマンドは、invoke 復元メッセージと metareplace コマンドによって実行する適切な対策を表示します。このコマンドは -f フラグを付けて実行する必要があります。この状態は、複数のスライスの障害のため、データが改ざんされた可能性があることを示しています。

RAID-5 ボリューム内のスライスの交換と有効化の概要

Solaris Volume Manager には、ミラーと RAID-5 ボリューム内のコンポーネントを交換し、有効化する機能があります。それを行なう場合の問題と要件は、ミラーと RAID-5 ボリュームで同じです。詳細については、248 ページの「RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要」を参照してください。

シナリオ-RAID-5ボリューム

RAID5 ボリュームでは、データの冗長性を得るために、2倍のストレージ合計容量を必要とする RAID1 ボリュームのオーバーヘッドなく、冗長ストレージを使用できます。RAID5 ボリュームを設定することで、同じ一連のディスクコンポーネントで、RAID-1 ボリュームによって実現可能な容量より大きな冗長ストレージを提供できます。さらに、ホットスペアを使用する(第16章「ホットスペアプール(概要)」および特に189ページの「ホットスペアのしくみ」を参照)と、ほぼ同じレベルの安全性が得られます。短所は、書き込み時間の増加とコンポーネント障害時の大幅なパフォーマンス低下です。多くの状況で、それらのトレードオフは重大ではありません。次の例では、第5章「Solaris Volume Managerの構成と使用(シナリオ)」で説明しているサンプルシナリオを利用して、RAID-5 ボリュームによって追加のストレージ容量を提供する方法を示します。

RAID-0 と RAID-1 ボリュームのその他シナリオでは、2つのコントローラに分散した6つのディスク上の6つのスライス(c1t1d0、c1t2d0、c1t3d0、c2t1d0、c2t2d0、c2t3d0)を使用して、27Gバイトの冗長ストレージが得られました。RAID-5 構成では、同じスライスを使用することによって45Gバイトのストレージが得られます。さらに、この構成では、1つのコンポーネントに障害が発生しても、データが失われたり、アクセスが中断されることはありません。この構成にホットスペアを追加することによって、RAID-5 ボリュームは追加のコンポーネントの障害に耐えることができます。このアプローチのもっとも重大な短所は、コントローラの障害によって、この RAID-5 ボリュームのデータが失われることです。110ページの「シナリオ-RAID-1 ボリューム(ミラー)」で説明する RAID-1 ボリュームでは、この問題は起こりません。

◆◆◆ 第 15 章

RAID-5 ボリューム (タスク)

この章では、RAID-5 ボリュームに関連する Solaris Volume Manager のタスクの実行について説明します。これらのタスクに関連する概念については、第 14 章「RAID-5 ボリューム (概要)」を参照してください。

RAID-5 ボリューム (タスクマップ)

次のタスクマップは、Solaris Volume Manager の RAID-5 ボリュームを管理するために必要な手順を示します。

タスク	説明	参照先
RAID-5 ボリュームを作成する	Solaris Volume Manager の GUI または <code>metainit</code> コマンドを使用して、RAID-5 ボリュームを作成します。	178 ページの「RAID-5 ボリュームを作成する方法」
RAID-5 ボリュームのステータスをチェックする	Solaris Volume Manager の GUI または <code>metastat</code> コマンドを使用して、RAID-5 ボリュームのステータスをチェックします。	179 ページの「RAID-5 ボリュームのステータスをチェックする方法」
RAID-5 ボリュームを拡張する	Solaris Volume Manager の GUI または <code>metattach</code> コマンドを使用して、RAID-5 ボリュームを拡張します。	180 ページの「RAID-5 ボリュームを拡張する方法」
RAID-5 ボリューム内のスライスを有効にする	Solaris Volume Manager の GUI または <code>metareplace</code> コマンドを使用して、RAID-5 ボリューム内のスライスを有効にします。	182 ページの「RAID-5 ボリューム内のコンポーネントを有効にする方法」
RAID-5 ボリューム内のスライスを置き換える	Solaris Volume Manager の GUI または <code>metareplace</code> コマンドを使用して、RAID-5 ボリューム内のスライスを置き換えます。	183 ページの「RAID-5 ボリューム内のコンポーネントを置き換える方法」

RAID-5 ボリュームの作成



注意 - 32ビットカーネルの Solaris ソフトウェアを実行する予定がある場合、または Solaris 9 4/03 リリースより前のバージョンの Solaris OS を使用する予定がある場合は、1T バイトを超えるボリュームを作成しないでください。Solaris Volume Manager でサポートされる大容量ボリュームの詳細については、52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」を参照してください。

▼ RAID-5 ボリュームを作成する方法

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」および 171 ページの「RAID-5 ボリュームを作成するための背景情報」を確認してください。

- RAID-5 ボリュームを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。「アクション」、「ボリュームを作成」の順に選択します。さらに、ウィザードの手順に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用します。

```
# metainit volume-name -r component component component -i interlace-value
```

`volume-name` 作成するボリュームの名前を指定します。

`-r` RAID-5 ボリュームを作成することを指定します。

`component` RAID-5 ボリュームに含めるスライスまたはソフトパーティションを指定します。少なくとも3つのコンポーネントが必要です。

`-i` 飛び越し値を指定します。

詳細は、`metainit(1m)` のマニュアルページを参照してください。

例 15-1 3つのスライスから成る RAID-5 ボリュームを作成する

この例では、`-r` オプションを使用して、3つのスライスから RAID-5 ボリューム `d45` を作成します。飛び越し値が指定されていないため、`d45` ではデフォルトの 16K バイトが使用されます。RAID-5 ボリュームが設定されたことを示すメッセージが表示され、ボリュームの初期化が開始されます。

ボリュームの初期化が終わるまで RAID-5 ボリュームは使用できません。

```
# metainit d45 -r c2t3d0s2 c3t0d0s2 c4t0d0s2
d45: RAID is setup
```

参照 新たに作成した RAID-5 ボリュームをファイルシステム用に準備する方法については、『Oracle Solaris の管理: デバイスとファイルシステム』の第 15 章「ファイルシステムの作成およびマウント(タスク)」を参照してください。データベースなどの一部のアプリケーションは、ファイルシステムを使用しません。これらのアプリケーションは、代わりに raw ボリュームを使用します。アプリケーションは独自の方法でボリュームを認識できなければなりません。

ホットスペアプールと RAID-5 ボリュームを関連付ける手順については、197 ページの「ホットスペアプールをボリュームに関連付ける方法」を参照してください。

RAID-5 ボリュームの保守

▼ RAID-5 ボリュームのステータスをチェックする方法

RAID-5 ボリュームのステータスをチェックする際、ボリュームの状態を完全に理解するには、RAID-5 の状態とスライスの状態の両方をチェックする必要があります。さらに、ボリュームの状態が「正常」でない場合は、データが失われた可能性についても知る必要があります。詳細は、172 ページの「RAID-5 ボリュームのステータスのチェックの概要」を参照してください。

注 - RAID-5 ボリュームの初期化や再同期を中断することはできません。

- RAID-5 ボリュームのステータスをチェックするには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、ボリュームのステータスを表示します。ボリュームを選択します。次に、「アクション」、「プロパティ」の順に選択して、詳細情報を確認します。詳細は、オンラインヘルプを参照してください。
 - `metastat` コマンドを使用して、RAID-5 ボリュームのステータスを表示します。

```
# metastat [-s diskset] [volume]
-s diskset   コマンドの実行対象となるディスクセットの名前を指定します。
volume      表示するボリュームを指定します。
```

metastat コマンドは、RAID-5 ボリュームのスライスごとに次の情報を表示します。

デバイス	ストライプ内のスライスのデバイス名を示します。
開始ブロック	スライスが開始するブロックを示します。
Dbase	スライスに状態データベースの複製が含まれているかどうかを示します
状態	スライスの状態を示します。
ホットスペア	スライスが障害スライスのホットスペアとして使用されているかどうかを示します

例 15-2 RAID-5 ボリュームのステータスを表示する

次の例は、RAID-5 ボリュームに関する metastat コマンドの出力です。

```
# metastat d10
d10: RAID
  State: Okay
  Interlace: 32 blocks
  Size: 10080 blocks
Original device:
  Size: 10496 blocks
  Device          Start Block  Dbase State      Hot Spare
  c0t0d0s1        330         No  Okay
  c1t2d0s1        330         No  Okay
  c2t3d0s1        330         No  Okay
```

metastat コマンドの出力には、ボリュームが RAID-5 ボリュームであることが示されています。この情報は、ボリューム名の後ろの「RAID」で識別できます。RAID-5 ボリューム内のスライスごとに、次のような出力が表示されます。

- ストライプ内のスライスの名前。
- スライスが開始するブロック。
- これらのスライスのどれにも状態データベースの複製が含まれていないという表示。
- スライスの状態。この例では、すべてのスライスが「正常」状態です。
- スライスが障害スライスのホットスペアかどうか。

▼ RAID-5 ボリュームを拡張する方法

一般に、コンポーネントの追加は、領域が不足している RAID-5 ボリュームに対する一時的な解決策です。パフォーマンス上の理由から、「純粋な」RAID-5 ボリュームの使用をお勧めします。ストレージ領域を増やすために既存の RAID-5 ボリュームを拡張する必要がある場合は、この手順を使用してください。



注意 - 32 ビットカーネルの Solaris ソフトウェアを実行する予定がある場合、または Solaris 9 4/03 リリースより前のバージョンの Solaris OS を使用する予定がある場合は、1T バイトを超えるボリュームを作成しないでください。Solaris Volume Manager でのマルチテラバイトボリュームのサポートの詳細は、52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」を参照してください。

始める前に 171 ページの「RAID-5 ボリュームを作成するための背景情報」を読んでください。

- 1 すべてのデータの最新のバックアップを取っていること、およびスーパーユーザーのアクセス権を持っていることを確認します。
- 2 RAID-5 ボリュームに追加のコンポーネントを接続するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、次に RAID-5 ボリュームを開きます。「コンポーネント」ペインを選択します。次に、「コンポーネントを割り当て」を選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metattach` コマンドを使用します。

```
# metattach volume-name name-of-component-to-add
```

`volume-name` 拡張する RAID-5 ボリュームの名前を指定します。

`name-of-component-to-add` RAID-5 ボリュームに接続するコンポーネントの名前を指定します。

詳細は、`metattach(1M)` のマニュアルページを参照してください。

例 15-3 RAID-5 ボリュームにコンポーネントを追加する

次の例では、スライス `c2t1d0s2` を既存の RAID-5 ボリューム `d2` に追加しています。

```
# metattach d2 c2t1d0s2
d2: column is attached
```

参照 UFS ファイルシステムの場合は、RAID-5 ボリュームに対して `growfs` コマンドを実行します。46 ページの「`growfs` コマンドを使用したボリュームおよびディスク容量の拡張」を参照してください。

データベースなどの一部のアプリケーションは、ファイルシステムを使用しません。これらのアプリケーションは、代わりに raw ボリュームを使用します。このような場合、アプリケーションは独自の方法で追加領域を拡張できなければなりません。

▼ RAID-5 ボリューム内のコンポーネントを有効にする方法

ディスクドライブに障害が発生した場合は、そのドライブの代わりにシステム上のほかのディスク (およびそのスライス) を使用することができます (183 ページの「RAID-5 ボリューム内のコンポーネントを置き換える方法」を参照)。または、ディスクを修復し、ラベルを付け、`-e` オプションを指定して `metareplace` コマンドを実行することによって、ディスクをもう一度有効化することもできます。

- 1 すべてのデータの最新のバックアップを取っていること、およびスーパーユーザーのアクセス権を持っていることを確認します。
- 2 RAID-5 ボリューム内の障害が発生したコンポーネントを有効にするには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから「ボリューム」ノードを開き、次に RAID-5 ボリュームを開きます。「コンポーネント」ペインを選択します。次に、障害が発生したコンポーネントを選択します。「コンポーネントを有効にする」をクリックします。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metareplace` コマンドを使用します。

```
# metareplace -e volume-name component-name
```

`-e` 障害が発生したコンポーネントを使用可能な状態に戻し、そのコンポーネントに対して再同期を実行することを指定します。

`volume-name` 障害コンポーネントが含まれているボリュームの名前を指定します。

`component-name` 障害が発生したコンポーネントの名前を指定します。

`metareplace` コマンドは、新しいコンポーネントと RAID-5 ボリュームの残りのコンポーネントとの再同期を自動的に開始します。

例 15-4 RAID-5 ボリューム内のコンポーネントを有効にする

次の例では、RAID-5 ボリューム `d20` 内のスライス `c2t0d0s2` にソフトエラーがあります。`metareplace` コマンドに `-e` オプションを指定して、このスライスを有効にします。

```
# metareplace -e d20 c2t0d0s2
```

▼ RAID-5 ボリューム内のコンポーネントを置き換える方法

このタスクでは、RAID-5 ボリュームの1つのスライスだけに障害が発生している場合に、そのスライスを置換します。



注意 - 複数のスライスでエラーが発生している状態で、障害のあるスライスを1つだけ置換すると、不正なデータが生成されることがあります。この場合、データの整合性に疑問が生じます。

metareplace コマンドを障害が発生していないデバイス上で実行すれば、ディスクスライスなどのコンポーネントを置換できます。この方法は、RAID-5 ボリュームのパフォーマンスを調整するときなどに便利です。

- 1 すべてのデータの最新のバックアップを取っていること、およびスーパーユーザーのアクセス権を持っていることを確認します。
- 2 次のいずれかの方法を使用して、RAID-5 ボリュームのどのスライスを置換する必要があるかを調べます。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。次に、RAID-5 ボリュームを開きます。「コンポーネント」ペインを選択します。個々のコンポーネントのステータスを表示します。詳細は、オンラインヘルプを参照してください。
 - metastat コマンドを使用します。


```
# metastat volume
volume    RAID-5 ボリュームの名前を指定します。
```

「保守が必要」というキーワードを探して、RAID-5 ボリュームの状態を調べます。「保守」というキーワードを探して、障害のあるスライスを特定します。
- 3 次のいずれかの方法を使用して、障害のあるスライスを別のスライスで置き換えます。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。次に、RAID-5 ボリュームを開きます。「コンポーネント」ペインを選択します。障害が発生したコンポーネントを選択します。「コンポーネントを置換」をクリックし、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の metareplace コマンドを使用します。

```
# metareplace volume-name failed-component new-component
```

- *volume-name* は、障害コンポーネントが含まれている RAID-5 ボリュームの名前です。
- *failed-component* は、置き換えられるコンポーネントの名前です。
- *new-component* は、障害のあるコンポーネントの代わりにボリュームに追加するコンポーネントの名前です。

詳細は、[metareplace\(1M\)](#) のマニュアルページを参照してください。

- 4 置換用スライスのステータスを確認するには、[手順 2](#) のいずれかの方法を使用します。

置換されたスライスの状態は、「再同期中」または「正常」になるはずです。

例 15-5 RAID-5 コンポーネントを置き換える

```
# metastat d1
d1: RAID
State: Needs Maintenance
  Invoke: metareplace d1 c0t14d0s6 <new device>
  Interlace: 32 blocks
  Size: 8087040 blocks
Original device:
  Size: 8087520 blocks
Device          Start Block  Dbase State      Hot Spare
c0t9d0s6        330         No   Okay
c0t13d0s6       330         No   Okay
c0t10d0s6       330         No   Okay
c0t11d0s6       330         No   Okay
c0t12d0s6       330         No   Okay
c0t14d0s6       330         No   Maintenance

# metareplace d1 c0t14d0s6 c0t4d0s6
d1: device c0t14d0s6 is replaced with c0t4d0s6
# metastat d1
d1: RAID
  State: Resyncing
  Resync in progress: 98% done
  Interlace: 32 blocks
  Size: 8087040 blocks
Original device:
  Size: 8087520 blocks
Device          Start Block  Dbase State      Hot Spare
c0t9d0s6        330         No   Okay
c0t13d0s6       330         No   Okay
c0t10d0s6       330         No   Okay
c0t11d0s6       330         No   Okay
c0t12d0s6       330         No   Okay
c0t4d0s6        330         No   Resyncing
```


この例では、`metastat` コマンドで、RAID-5 ボリューム `d1` 内の障害の発生したスライスを表示します。使用可能なスライスを特定してから `metareplace` コマンドを実行します。このコマンドには、まず障害が発生したスライスを指定し、次に置換用のスライスを指定します。

使用可能なスライスがない場合は、`metareplace` コマンドに `-e` オプションを付けて実行し、障害のあるデバイスを再同期することによって、予想されるソフトエラーからの回復を試みます。この手順の詳細については、[182 ページの「RAID-5 ボリューム内のコンポーネントを有効にする方法」](#)を参照してください。複数のエラーがある場合は、まず「保守」状態のスライスを置換するか有効にする必要があります。そのあとで、「最後にエラー」状態のスライスを修復します。`metareplace` コマンドの実行後、`metastat` コマンドを使用すると、再同期の進捗状況をモニターできます。置換中は、ボリュームと新しいスライスは「再同期中」状態となります。この状態の間は、ボリュームを使い続けることができます。

ホットスペアプール(概要)

この章では、Solaris Volume Manager がホットスペアプールをどのように使用するかを説明します。関連タスクの実行については、第 17 章「ホットスペアプール(タスク)」を参照してください。

この章では、次の内容について説明します。

- 187 ページの「ホットスペアとホットスペアプールの概要」
- 191 ページの「シナリオ-ホットスペア」

ホットスペアとホットスペアプールの概要

「ホットスペアプール」はスライス(「ホットスペア」)のコレクションで、Solaris Volume Manager が RAID-1 (ミラー) および RAID-5 ボリュームのデータ可用性を向上させるために使用します。スライスで障害が発生すると、サブミラーまたは RAID-5 ボリュームで、Solaris Volume Manager が障害の発生したスライスをホットスペアに自動的に置換します。

注-ホットスペアは RAID-0 ボリュームまたは一方向のミラーには適用されません。自動置換が機能するには、冗長データが使用可能である必要があります。

ホットスペアは、アイドル時に、データや状態データベースの複製を保持するために使用できません。ホットスペアは、それに関連付けられているボリュームでスライスの障害が発生した場合に、すぐに使用できるように準備しておく必要があります。ホットスペアを使用するには、システムが実際に機能するために必要なディスク以上に追加のディスクに投資する必要があります。

Solaris Volume Manager により、ホットスペアプール内のホットスペアを動的に追加、削除、置換、有効にすることができます。Solaris 管理コンソールまたはコマンド

ド行ユーティリティーを使用して、ホットスペアおよびホットスペアプールを管理できます。これらのタスクの詳細については、第17章「ホットスペアプール(タスク)」を参照してください。

ホットスペア

ホットスペアとは、機能し、使用可能ですが、使用中でないスライス(ボリュームでない)です。ホットスペアは予約されています。つまり、サブミラーまたはRAID-5ボリューム内の障害の発生したスライスの置換に備えて待機しています。

ホットスペアはハードウェアの障害から保護します。RAID-1 および RAID-5 ボリュームのスライスは、障害の発生時にホットスペアによって自動的に置換されます。ホットスペアは、ボリュームで使用できるように再同期化されます。ホットスペアは、障害の発生したサブミラーや RAID-5 ボリュームスライスが修正されるか、交換されるまで、一時的に使用することができます。

ホットスペアプール内にホットスペアを作成します。個々のホットスペアは1つまたは複数のホットスペアプールに含めることができます。たとえば、2つのサブミラーと2つのホットスペアがあるとしめます。ホットスペアは、各プールに優先順位の異なる2つのホットスペアがある2つのホットスペアプールとして配置できます。この方法により、最初に使用されるホットスペアを指定できます。この方法は、使用可能なホットスペアを増やすことで、可用性も向上します。

サブミラーまたは RAID-5 ボリュームは、サブミラーまたは RAID-5 ボリューム内の障害の発生したスライスのサイズ以上のホットスペアのみを使用できます。たとえば、1Gバイトのドライブから構成されたサブミラーがある場合、サブミラーのホットスペアは1Gバイト以上である必要があります。

ホットスペアプール

ホットスペアプールは、ホットスペアの順序付きリスト(コレクション)です。

最少のスライスから、最大の柔軟性と保護を得るために、1つまたは複数のホットスペアプールにホットスペアを配置できます。ホットスペアプールとして使用するよう指定した単一のスライスを、それぞれ異なるスライスと特性を持つ、複数のホットスペアプールに入れることができます。次に、ホットスペアプールを任意の数のサブミラーボリュームまたは RAID-5 ボリュームに割り当てることができます。

注-単一のホットスペアプールを複数のサブミラーまたは RAID-5 ボリュームに割り当てることができます。ただし、サブミラーまたは RAID-5 ボリュームに関連付けることができるホットスペアプールは1つだけです。

ホットスペアのしくみ

I/O エラーが発生すると、Solaris Volume Manager は、ホットスペアがホットスペアプールに追加された順番に基づいて、ホットスペアプールでホットスペアを検索します。Solaris Volume Manager は、置換対象のスライスのサイズ以上で、最初に使用可能なホットスペアについて、ホットスペアプールをチェックします。十分な大きさのある、Solaris Volume Manager によって最初に見つけられたホットスペアが置換先として使用されます。Solaris Volume Manager はホットスペアのステータスを「使用中」にし、必要に応じて、データを自動的に再同期化します。置換が行なわれても、ホットスペアプール内のホットスペアの順番は変更されません。

ミラーの場合、ホットスペアは機能するサブミラーからのデータで再同期化されます。RAID-5 ボリュームの場合、ホットスペアは、ボリューム内の他のスライスによって再同期化されます。ホットスペアのリストに、十分なサイズのスライスが見つからない場合、障害の発生したサブミラーまたは RAID-5 ボリュームは失敗状態になり、ホットスペアは使用されないままになります。サブミラーの場合、サブミラーはデータを完全に複製しなくなります。RAID-5 ボリュームの場合、データの冗長性は使用できなくなります。

ヒント-ホットスペアプールにホットスペアを追加する場合、最小から最大のサイズでそれらを追加します。この方法により、小さいスライスの置換先として、「大きな」ホットスペアを無駄にする可能性を避けます。

スライスで I/O エラーが発生すると、障害の発生したスライスが「障害」状態に置かれます。この状況を修正するには、まず障害の発生したスライスを修復するか、交換します。次に、Solaris 管理コンソール内の拡張ストレージツールを使用して、スライスを「使用可能」状態に戻します。または、`metahs -e` コマンドを使用します。

サブミラーまたは RAID-5 ボリュームは、障害の発生したスライスが有効になるか、交換されるまで、障害の発生したスライスの代わりにホットスペアを使用します。その後、ホットスペアは、ホットスペアプール内で「使用可能」とマークされます。このホットスペアは再び使用の準備が整えられます。

ホットスペアプールの状態

次の表に、ホットスペアプールの状態と実行可能なアクションについて説明します。

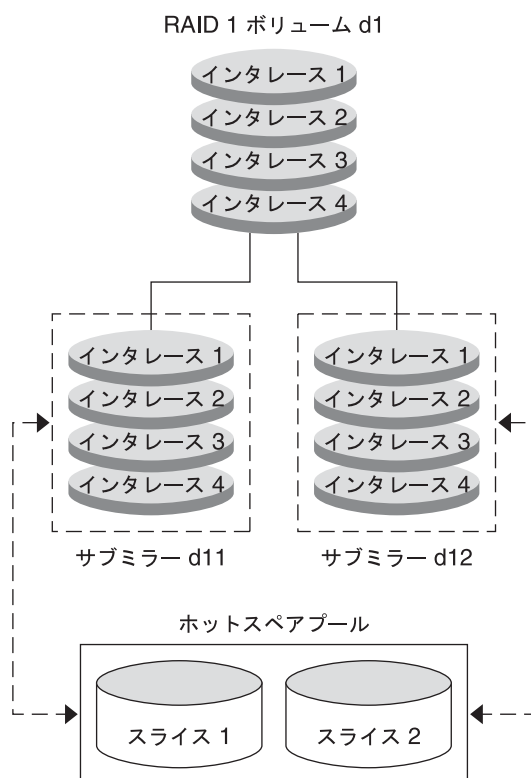
表 16-1 ホットスペアプールの状態(コマンド行)

状態	意味	アクション
使用可能	ホットスペアプール内のホットスペアは実行中で、データを受け付ける準備ができています。ホットスペアは現在書き込み中または読み取り中ではありません。	なし。
使用中	このホットスペアプールには、冗長ボリューム内の障害の発生したスライスに置き換わる使用中のホットスペアが含まれています。	ホットスペアの使用状態を診断します。次に、ホットスペアが使用されているボリューム内のスライスを修復します。
障害	ホットスペアまたはホットスペアプールに問題があります。ただし、ただちにデータを失う危険性はありません。このステータスは、すべてのホットスペアが使用中の場合にも表示されます。	ホットスペアの使用の状態またはそれらが破損している理由を診断します。必要に応じて、ホットスペアプールにさらにホットスペアを追加できます。

例 — ホットスペアプール

図 16-1 に、ミラー d1 内のサブミラー d11 および d12 に関連付けられているホットスペアプールを示します。いずれかのサブミラー内のスライスで障害が発生した場合、自動的にホットスペアが、障害の発生したスライスに置換されます。ホットスペアプール自体は、ミラーではなく、各サブミラーボリュームに関連付けられません。ホットスペアプールは、必要に応じて、他のサブミラーまたは RAID-5 ボリュームに関連付けることもできます。

図 16-1 ホットスペアプールの例



シナリオ-ホットスペア

ホットスペアは、データの損失に対して保護するために、冗長ボリューム (RAID-1 および RAID-5) に追加の保護を提供します。ホットスペアを、RAID-0 サブミラーまたは RAID-5 構成から構成される基盤のスライスに関連付けることにより、障害の発生したスライスを、ホットスペアプールの動作するスペアで自動的に置換させることができます。使用するように切り替えられたそれらのスライスは、必要な情報で更新されます。それにより、スライスは元のスライスとまったく同じように機能し続けることができます。障害の発生したスライスは、都合に応じて交換できます。

◆◆◆ 第 17 章

ホットスペアプール(タスク)

この章では、Solaris Volume Manager のホットスペアとホットスペアプールを操作する方法について説明します。関連する概念については、第 16 章「ホットスペアプール(概要)」を参照してください。

ホットスペアプール(タスクマップ)

次のタスクマップに、Solaris Volume Manager のホットスペアプールを管理するために必要な手順を示します。

タスク	説明	参照先
ホットスペアプールを作成する	ホットスペアプールを作成するには、Solaris Volume Manager GUI または <code>metainit</code> コマンドを使用します。	194 ページの「ホットスペアプールを作成する方法」
ホットスペアプールにスライスを追加する	ホットスペアプールにスライスを追加するには、Solaris Volume Manager GUI または <code>metahs</code> コマンドを使用します。	195 ページの「ホットスペアプールにスライスを追加する方法」
ホットスペアプールをボリュームに関連付ける	ホットスペアプールをボリュームに関連付けるには、Solaris Volume Manager GUI または <code>metaparam</code> コマンドを使用します。	197 ページの「ホットスペアプールをボリュームに関連付ける方法」
ボリュームに関連付けられているホットスペアプールを変更する	ボリュームに関連付けられているホットスペアプールを変更するには、Solaris Volume Manager GUI または <code>metaparam</code> コマンドを使用します。	198 ページの「関連付けられているホットスペアプールを変更する方法」

タスク	説明	参照先
ホットスペアとホットスペアプールのステータスを確認する	ホットスペアまたはホットスペアプールのステータスを確認するには、Solaris Volume Manager GUI、metastat コマンド、または metahs -i コマンドを使用します。	200 ページの「ホットスペアとホットスペアプールのステータスを確認する方法」
ホットスペアプール内のホットスペアを置き換える	ホットスペアプール内のホットスペアを置き換えるには、Solaris Volume Manager GUI または metahs コマンドを使用します。	200 ページの「ホットスペアプール内のホットスペアを置き換える方法」
ホットスペアプールからホットスペアを削除する	ホットスペアプールからホットスペアを削除するには、Solaris Volume Manager GUI または metahs コマンドを使用します。	202 ページの「ホットスペアプールからホットスペアを削除する方法」
ホットスペアを有効にする	ホットスペアプール内のホットスペアを有効にするには、Solaris Volume Manager GUI または metahs コマンドを使用します。	203 ページの「ホットスペアを有効にする方法」

ホットスペアプールの作成

▼ ホットスペアプールを作成する方法



注意 - 32 ビットカーネルで Solaris ソフトウェアを実行することを予定している場合や、Solaris 9 4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、1T バイトを超えるボリュームまたはホットスペアを作成しないでください。Solaris Volume Manager でのマルチテラバイトボリュームのサポートの詳細は、[52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」](#)を参照してください。



注意 - 十分な大きさが無いホットスペアを作成しても、Solaris Volume Manager から警告は表示されません。ホットスペアのサイズが接続先のボリュームのサイズ以上でない場合、そのホットスペアは機能しません。

始める前に [51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」](#)を確認してください。

- 1 スーパーユーザーになります。

- 2 ホットスペアプールを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開きます。次に、「アクション」、「ホットスペアプールの作成」の順に選択します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metainit` コマンドを使用します。

```
# metainit hot-spare-pool-name ctds-for-slice
```

`hot-spare-pool-name` ホットスペアプールの名前を指定します。

`ctds-for-slice` ホットスペアプールに追加されるスライス指定します。このオプションは、ホットスペアプールに追加されるスライスごとに繰り返されます。

詳細については、`metainit(1m)` のマニュアルページを参照してください。

注-ホットスペアプールはまた、`metahs` コマンドでも作成できます。

例 17-1 ホットスペアプールの作成

```
# metainit hsp001 c2t2d0s2 c3t2d0s2
hsp001: Hotspare pool is setup
```

この例では、ホットスペアプール `hsp001` にホットスペアとして 2 つのディスクが含まれています。システムは、ホットスペアプールが設定されていることを確認します。

参照 ホットスペアプールにさらに多くのホットスペアを追加するには、[195 ページ](#)の「[ホットスペアプールにスライスを追加する方法](#)」を参照してください。ホットスペアプールを作成したあと、それをサブミラーまたは RAID-5 ボリュームに関連付ける必要があります。[197 ページ](#)の「[ホットスペアプールをボリュームに関連付ける方法](#)」を参照してください。

▼ ホットスペアプールにスライスを追加する方法

始める前に [51 ページ](#)の「[Solaris Volume Manager コンポーネントを作成するための前提条件](#)」を確認してください。

- 1 スーパーユーザーになります。
- 2 既存のホットスペアプールにスライスを追加するには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開きます。変更するホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペア」パネルを選択します。詳細は、オンラインヘルプを参照してください。
- 次のいずれかの形式の `metahs` コマンドを使用します。

```
# metahs -a hot-spare-pool-name slice-to-add
```

```
# metahs -a -all hot-spare-pool-name slice-to-add
```

`-a hot-spare-pool-name` 指定されたホットスペアプールにスライスを追加するように指定します。

`-a all` すべてのホットスペアプールにスライスを追加するように指定します。

`slice-to-add` ホットスペアプールに追加するスライスを指定します。

詳細は、[metahs\(1M\)](#) のマニュアルページを参照してください。

注-ホットスペアを1つまたは複数のホットスペアプールに追加できます。ホットスペアをホットスペアプールに追加すると、そのホットスペアは、ホットスペアプール内のスライスのリストの最後に追加されます。

例 17-2 1つのホットスペアプールへのホットスペアスライスの追加

この例では、`-a` オプションにより、スライス `/dev/dsk/c3t0d0s2` がホットスペアプール `hsp001` に追加されます。システムは、スライスがホットスペアプールに追加されていることを確認します。

```
# metahs -a hsp001 /dev/dsk/c3t0d0s2
hsp001: Hotspare is added
```

例 17-3 すべてのホットスペアプールへのホットスペアスライスの追加

この例では、`all` とともに使用されている `-a` オプションにより、スライス `/dev/dsk/c3t0d0s2` がシステム上に構成されているすべてのホットスペアプールに追加されます。システムは、スライスがすべてのホットスペアプールに追加されていることを確認します。

```
# metahs -a -all /dev/dsk/c3t0d0s2
hsp001: Hotspare is added
hsp002: Hotspare is added
hsp003: Hotspare is added
```

ホットスペアプールのボリュームへの関連付け

▼ ホットスペアプールをボリュームに関連付ける方法

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」を確認してください。

- 1 スーパーユーザーになります。
- 2 ホットスペアプールを RAID-5 ボリュームまたはサブミラーに関連付けるには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」を開き、ボリュームを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペアプール」パネルを選択します。最後に、「HSP を割り当て」を選択します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metaparam` コマンドを使用します。

```
# metaparam -h hot-spare-pool component
-h                指定されたホットスペアプールを変更するように指定します。
hot-spare-pool   ホットスペアプールの名前を指定します。
component        ホットスペアプールを関連付ける先のサブミラーまたは RAID-5
                  ボリュームの名前を指定します。
```

詳細は、[metaparam\(1M\)](#) のマニュアルページを参照してください。

例 17-4 ホットスペアプールのサブミラーへの関連付け

次の例では、`-h` オプションにより、ホットスペアプール `hsp100` がミラー `d0` の 2 つのサブミラー `d10` と `d11` に関連付けられます。`metastat` コマンドは、このホットスペアプールがサブミラーに関連付けられていることを示しています。

```
# metaparam -h hsp100 d10
# metaparam -h hsp100 d11
# metastat d0
d0: Mirror
   Submirror 0: d10
      State: Okay
   Submirror 1: d11
      State: Okay
...
```

```
d10: Submirror of d0
    State: Okay
    Hot spare pool: hsp100
...

d11: Submirror of d0
    State: Okay
    Hot spare pool: hsp100
...
```

例 17-5 ホットスペアプールの RAID-5 ボリュームへの関連付け

次の例では、`-h` オプションにより、ホットスペア `hsp001` が RAID-5 ボリューム `d10` に関連付けられます。`metastat` コマンドは、このホットスペアプールが RAID-5 ボリュームに関連付けられていることを示しています。

```
# metaparam -h hsp001 d10
# metastat d10
d10: RAID
    State: Okay
    Hot spare pool: hsp001
...
```

▼ 関連付けられているホットスペアプールを変更する方法

始める前に [51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」](#)を確認してください。

- 1 スーパーユーザーになります。
- 2 ボリュームに関連付けられているホットスペアプールを変更するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開き、ボリュームを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペアプール」パネルを選択します。画面の指示に従うことによって、不要なホットスペアプールを切り離し、新しいホットスペアプールを接続します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metaparam` コマンドを使用します。

```
# metaparam -h hot-spare-pool-name RAID5-volume-or-submirror-name
-h          指定されたホットスペアプールを変更するように指定します。
```

<i>hot-spare-pool</i>	新しいホットスペアプールの名前を指定するか、または特殊なキーワード <code>none</code> を指定してホットスペアプールの関連付けを削除します。
<i>component</i>	ホットスペアプールを接続する先のサブミラーまたは RAID-5 ボリュームの名前を指定します。

詳細は、[metaparam\(1M\)](#) のマニュアルページを参照してください。

例 17-6 ホットスペアプールの関連付けの変更

次の例では、ホットスペアプール `hsp001` が、最初に RAID-5 ボリューム `d4` に関連付けられています。次に、ホットスペアプールのボリュームへの関連付けが `hsp002` に変更されます。`metastat` コマンドは、この変更の前とあとのホットスペアプールの関連付けを表示します。

```
# metastat d4
d4: RAID
   State: Okay
   Hot spare pool: hsp001
...
# metaparam -h hsp002 d4
# metastat d4
d4: RAID
   State: Okay
   Hot spare pool: hsp002
...
```

例 17-7 ボリュームのホットスペアプールの関連付けの削除

次の例では、ホットスペアプール `hsp001` が、最初に RAID-5 ボリューム `d4` に関連付けられています。次に、ホットスペアプールの関連付けが `none` に変更されます。このパラメータは、このボリュームにホットスペアプールを関連付けないようにすることを示します。`metastat` コマンドは、この変更の前とあとのホットスペアプールの関連付けを表示します。

```
# metastat d4
d4: RAID
   State: Okay
   Hot spare pool: hsp001
...
# metaparam -h none d4
# metastat d4
d4: RAID
   State: Okay
   Hot spare pool:
...
```

ホットスペアプールの保守

以降のセクションでは、ホットスペアプールに対する保守タスクを実行する方法を示します。

▼ ホットスペアとホットスペアプールのステータスを確認する方法

- ホットスペアプールとそのホットスペアのステータスを表示するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開き、ホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択して、詳細なステータス情報を表示します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metastat` コマンドを実行します。

```
# metastat hot-spare-pool-name
```

例 17-8 ホットスペアプールのステータスの表示

次の例は、ホットスペアプールに対する `metastat` コマンドからの出力例を示しています。

```
# metastat hsp001
hsp001: 1 hot spare
      c1t3d0s2           Available      16800 blocks
```

ホットスペアプールのステータスはまた、`metahs` コマンドでも確認できます。

ホットスペアプールの状態と、その考えられる対処方法については、[189 ページの「ホットスペアプールの状態」](#)を参照してください。

▼ ホットスペアプール内のホットスペアを置き換える方法

- 1 スーパーユーザーになります。
- 2 次のいずれかの方法を使用して、ホットスペアが現在使用されているかどうかを確認します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開き、ホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペア」パネルを選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metastat` コマンドを使用して、ホットスペアプールのステータスを表示します。

```
# metastat hot-spare-pool-name
```

詳細は、[metastat\(1M\)](#) のマニュアルページを参照してください。

3 ホットスペアを置き換えるには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開き、ホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペア」パネルを選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
- 次の形式の `metahs` コマンドを使用します。

```
# metahs -r hot-spare-pool-name current-hot-spare replacement-hot-spare
```

`-r` 指定されたホットスペアプール内のディスクを置き換えるように指定します。

`hot-spare-pool-name` ホットスペアプールの名前を指定します。また、特殊なキーワード `all` を使用して、ホットスペアプールのすべての関連付けを変更することもできます。

`current-hot-spare` 置き換えられる現在のホットスペアの名前を指定します。

`replacement-hot-spare` 指定されたホットスペアプール内の現在のホットスペアを置き換えるスライスの名前を指定します。

詳細は、[metahs\(1M\)](#) のマニュアルページを参照してください。

例 17-9 1つのホットスペアプール内のホットスペアの置き換え

次の例では、`metastat` コマンドは、このホットスペアが使用されていないことを示しています。`metahs -r` コマンドは、ホットスペアプール `hsp003` 内のホットスペア `/dev/dsk/c0t2d0s2` をホットスペア `/dev/dsk/c3t1d0s2` に置き換えます。

```
# metastat hsp003
hsp003: 1 hot spare
        c0t2d0s2          Broken          5600 blocks
# metahs -r hsp003 c0t2d0s2 c3t1d0s2
```

```
hsp003: Hotspare c0t2d0s2 is replaced with c3t1d0s2
```

例 17-10 関連付けられているすべてのホットスペアプール内のホットスペアの置き換え

次の例では、キーワード `all` により、ホットスペア `/dev/dsk/c1t0d0s2` が、それに関連付けられているすべてのホットスペアプール内でホットスペア `/dev/dsk/c3t1d0s2` に置き換えられます。

```
# metahs -r all c1t0d0s2 c3t1d0s2
hsp001: Hotspare c1t0d0s2 is replaced with c3t1d0s2
hsp002: Hotspare c1t0d0s2 is replaced with c3t1d0s2
hsp003: Hotspare c1t0d0s2 is replaced with c3t1d0s2
```

▼ ホットスペアプールからホットスペアを削除する方法

- 1 スーパーユーザーになります。
- 2 次のいずれかの方法を使用して、ホットスペアが現在使用されているかどうかを確認します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開き、ホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペア」パネルを選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metastat` コマンドを使用して、ホットスペアプールのステータスを表示します。

```
# metastat hot-spare-pool-name
```

[metastat\(1M\)](#) のマニュアルページを参照してください。

- 3 ホットスペアを削除するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開き、ホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペア」パネルを選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metahs` コマンドを使用します。

```
# metahs -d hot-spare-pool-name current-hot-spare
```

- d 指定されたホットスペアプールからホットスペアを削除するように指定します。
 - hot-spare-pool* ホットスペアプールの名前を指定します。また、特殊なキーワード `all` を使用して、すべてのホットスペアプールからホットスペアを削除することもできます。
 - current-hot-spare* 削除される現在のホットスペアの名前を指定します。
- 詳細は、[metahs\(1M\)](#) のマニュアルページを参照してください。

例 17-11 1つのホットスペアプールからのホットスペアの削除

次の例では、`metastat` コマンドは、このホットスペアが使用されていないことを示しています。`metahs -d` コマンドは、ホットスペアプール `hsp003` 内のホットスペア `/dev/dsk/c0t2d0s2` を削除します。

```
# metastat hsp003
hsp003: 1 hot spare
          c0t2d0s2          Broken          5600 blocks
# metahs -d hsp003 c0t2d0s2
```

▼ ホットスペアを有効にする方法

- 1 スーパーユーザーになります。
- 2 ホットスペアを「使用可能」状態に戻すには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ホットスペアプール」ノードを開き、ホットスペアプールを選択します。「アクション」、「プロパティ」の順に選択します。次に、「ホットスペア」パネルを選択します。画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metahs` コマンドを使用します。

```
# metahs -e hot-spare-slice
-e          ホットスペアを有効にするように指定します。
hot-spare-slice 有効にするスライスの名前を指定します。
```

詳細は、[metahs\(1M\)](#) のマニュアルページを参照してください。

例 17-12 ホットスペアの有効化

次の例では、`metahs` コマンドは、ホットスペア `/dev/dsk/c0t0d0s2` が修復されたあと、そのホットスペアを「使用可能」状態にします。ホットスペアプールを指定する必要はありません。

```
# metahs -e c0t0d0s2
```

ディスクセット (概要)

この章では、ディスクセットに関する概念について説明します。関連タスクの実行方法については、[第19章「ディスクセット\(タスク\)」](#)を参照してください。

この章の内容は次のとおりです。

- 205 ページの「ディスクセットの新機能」
- 205 ページの「ディスクセットの概要」
- 208 ページの「Solaris Volume Manager ディスクセット管理」
- 215 ページの「ディスクセットの操作のガイドライン」
- 216 ページの「ディスクセット内の非同期共有ストレージ」
- 216 ページの「シナリオ-ディスクセット」

ディスクセットの新機能

このセクションでは、この Solaris リリースの新しいディスクセット機能について説明します。

Solaris リリースの新機能のすべての一覧と説明については、『[Oracle Solaris 10 1/13 の新機能](#)』を参照してください。

ディスクセットの概要

ディスクセットは論理ボリュームとホットスペアを含む一連の物理ストレージボリュームです。ボリュームとホットスペアプールは、そのディスクセット内からのドライブ上に構築する必要があります。ディスクセット内にボリュームを作成したら、物理スライスと同じようにそのボリュームを使用できます。ボリュームを使用して、ファイルシステムを作成してマウントし、データを格納します。

注-ディスクセットは、SPARC ベースと x86 ベースの両方のプラットフォームでサポートされています。

ディスクセットのタイプ

このセクションでは、Solaris Volume Manager で使用可能なさまざまなディスクセットのタイプについて説明します。

ローカルディスクセット

各ホストには、ローカルディスクセットがあります。ローカルディスクセットは、名前付きディスクセットに含まれていない、ホスト上のすべてのディスクから構成されます。ローカルディスクセットは特定のホストに排他的に属します。ローカルディスクセットには、その特定のホスト構成の状態データベースが格納されます。ローカルディスクセット内のボリュームおよびホットスワッププールは、ローカルディスクセット内のドライブからのみ構成されます。

名前付きディスクセット

ローカルディスクセットに加えて、ホストは名前付きディスクセットに参加できません。名前付きディスクセットは、ローカルディスクセットに含まれないディスクセットです。システムの構成に応じて、次のタイプの名前付きディスクセットを実装して、ボリュームを管理できます。

共有ディスクセット

共有ディスクセットは複数のホストで共有できます。共有ディスクセットは、すべての参加しているホストから表示できますが、それにアクセスできるのは、ディスクセットの所有者だけです。各ホストは、共有ディスクセットを制御できますが、一度に制御できるのは1台のホストだけです。さらに、共有ディスクセットは、個別の名前空間を提供し、その中でボリュームが管理されます。

共有ディスクセットは、データの冗長性とデータの可用性をサポートします。1台のホストで障害が発生すると、別のホストが、障害のあるホストのディスクセットを引き継ぐことができます(このタイプの構成はフェイルオーバー構成と呼ばれます)。

注 - 共有ディスクセットは、Sun Cluster、Solstice HA (High Availability)、または他のサポートされるサードパーティー HA フレームワークで使用することも意図されています。Solaris Volume Manager は、単独ではフェイルオーバー構成を実装するために必要なすべての機能を提供しません。

各ホストはディスクのセットを制御できますが、一度に制御できるホストは1台のみです。

auto-take ディスクセット

Solaris 9 4/04 リリースで、auto-take 機能が使用できるようになるまで、Solaris Volume Manager では、`/etc/vfstab` ファイルによるディスクセット上のファイルシステムの自動マウントをサポートしていませんでした。Solaris Volume Manager では、ディスクセット上のファイルシステムにアクセスできるようにするために、システム管理者が `metaset -s setname -t` コマンドを使用して、ディスクセットの取得コマンドを手動で発行する必要がありました。

auto-take 機能により、`metaset -s setname -A enable` コマンドを使用して、ブート時に自動的にディスクセットにアクセスされるように設定できます。auto-take 機能により、ブート時に、`/etc/vfstab` ファイルにファイルシステムのマウントオプションを定義できます。この機能を使用して、有効なディスクセット内のボリューム上のファイルシステムのマウントオプションを `/etc/vfstab` ファイルに定義できます。

auto-take 機能をサポートするのは、単一ホストのディスクセットのみです。auto-take 機能では、ディスクセットが他のシステムと共有されていない必要があります。共有されているディスクセットは、auto-take 機能を使用するように設定できず、`metaset -A` コマンドは失敗します。ただし、ディスクセットから他のホストが削除されると、auto-take するように設定できます。同様に、auto-take ディスクセットには他のホストを追加できません。auto-take 機能を無効にしている場合は、他のホストをディスクセットに追加できます。

注 - Sun Cluster 環境では、auto-take 機能は無効です。Sun Cluster はディスクセットの取得と解放を処理します。

auto-take 機能の詳細については、`metaset(1M)` コマンドの `-A` オプションを参照してください。

複数所有者ディスクセット

Sun Cluster 環境に作成される名前付きディスクセットは、複数所有者ディスクセットと呼ばれます。複数所有者ディスクセットにより、複数のノードがディスクセットの所有権を共有し、共有ディスクに同時にアクセスできます。複数所有者

ディスクセット内のすべてのディスクとボリュームは、クラスタのすべてのノードによって直接アクセスできます。各複数所有者ディスクセットには、ディスクセットに追加されたホストのリストが格納されます。結果として、クラスタ構成内の各複数所有者ディスクセットには異なる(時には重複する)一連のホストを指定できます。

各複数所有者ディスクセットにはマスターノードがあります。マスターノードの機能は、状態データベースの複製の変更を管理し、更新することです。ディスクセットごとにマスターノードがあるため、複数のマスターが同時に存在できません。マスターが選択される方法は2つあります。最初の方法は、ディスクを最初にディスクセットに追加したノードがマスターになることです。2つ目の方法は、マスターノードでパニックが発生し、障害が発生した場合です。最小のノードIDを持つノードがマスターノードになります。

Sun Cluster 環境で、複数所有者ディスクセット機能は、複数所有者ディスクセットストレージを管理するためにのみ有効にされます。Solaris Volume Manager for Sun Cluster 機能は Sun Cluster 10/04 ソフトウェアコレクションから始まる Sun Cluster のリリースと Oracle Real Applications Clusters などのアプリケーションと連携します。Solaris Volume Manager for Sun Cluster の詳細については、第4章「Solaris Volume Manager for Sun Cluster (概要)」を参照してください。

複数所有者ディスクセットを構成する前に、Solaris OSに加えて、次のソフトウェアがインストールされている必要があります。

- Sun Cluster 初期クラスタフレームワーク
- Sun Cluster Support for Oracle Real Application Clusters ソフトウェア
- Oracle Real Application Clusters ソフトウェア

注 - Sun Cluster および Oracle Real Application Clusters ソフトウェアの設定については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』と『[Oracle Solaris Cluster Data Service for Oracle Real Application Clusters ガイド](#)』を参照してください。

Solaris Volume Manager ディスクセット管理

ローカルディスクセット管理と異なり、ディスクセット状態データベースを手動で作成または削除する必要はありません。Solaris Volume Manager は、ディスクセット内に 50 個までの複製の合計数で、ディスクセット内のすべてのディスクで、各ディスク上(スライス7上)に、1つの状態データベースの複製を配置します。

ディスクセットにディスクを追加すると、Solaris Volume Manager によって、ディスクセットに状態データベースの複製が自動的に作成されます。ディスクがディスクセットに受け入れられると、ディスクセットの状態データベースの複製をディスクに配置できるように、Solaris Volume Manager によってディスクが再分割されることがあります(211 ページの「[自動ディスクパーティション分割](#)」を参照)。

通常ディスクセット内のボリューム上に存在するファイルシステムは、`/etc/vfstab` ファイルによって、ブート時に自動的にマウントされません。必要な Solaris Volume Manager RPC デーモン (`rpc.metad` と `rpc.metamhd`) は、ブートプロセスでこれを許可するほど早く起動しません。さらに、リブート時にディスクセットの所有権が失われます。`/etc/inetd.conf` ファイルの Solaris Volume Manager RPC デーモンを無効にしないでください。それらはデフォルトで起動するように構成されています。Solaris Volume Manager がそのすべての機能を使用できるように、これらのデーモンは有効のままにしておく必要があります。

`metaset` コマンドの `-A` オプションを使用して、`auto-take` 機能が有効にされている場合、ブート時にディスクセットが自動的に取得されます。これらの環境において、ディスクセット内のボリュームに存在するファイルシステムは、`/etc/vfstab` ファイルによって自動的にマウントできます。ブートプロセス時に自動取得を有効にするには、ディスクセットが単一のホストのみに関連付けられており、`auto-take` 機能が有効にされている必要があります。ディスクセットは、ディスクセットの作成時または作成後に有効にできます。`auto-take` 機能の詳細については、[207 ページの「auto-take ディスクセット」](#)を参照してください。

注- ディスクセットは単一ホスト構成でサポートされますが、多くの場合「ローカル」(デュアル接続されていない) 使用には適していません。2つの一般的な例外は、論理ボリュームに管理しやすい名前空間を指定するか、記憶域ネットワーク (SAN) ファブリック上のストレージを簡単に管理するために、ディスクセットを使用することです ([216 ページの「シナリオ-ディスクセット」](#)を参照)。

ディスクセットを作成し、構成するには、Solaris Volume Manager コマンド行インタフェース (`metaset` コマンド) または Solaris 管理コンソール内の拡張ストレージツールを使用します。

ディスクセットにディスクが追加されたら、ディスクセット内のホストによって、ディスクを予約(または取得)および解放できます。ホストによってディスクが予約されると、ディスクセット内の他のホストはディスクセット内のディスク上のデータにアクセスできません。ディスクセットの保守を実行するには、ホストがディスクセットの所有者であるか、ディスクセットを予約している必要があります。ホストがディスクセットの暗黙的所有権を取得するには、最初のディスクをセットに入れます。

ディスクセットは、別のシステムに作成されたディスクセットも含めて、`metainport` コマンドを使用して、既存の Solaris Volume Manager 構成にインポートできます。

ディスクセットの予約

ホストはディスクセット内のディスクを使用する前に、ディスクセットを予約する必要があります。ディスクセットを予約するには2つの方法があります。

- 安全に - ディスクセットを安全に予約する場合、Solaris Volume Manager はディスクセットの取得を試み、他のホストはディスクセットの解放を試みます。解放（そのために、予約）は失敗することがあります。
- 強制的に - ディスクセットを強制的に予約する場合、Solaris Volume Manager は、別のホストが現在セットを予約しているかどうかにかかわらず、ディスクセットを予約します。この方法は、一般にディスクセット内のホストが停止しているか、通信していない場合に使用します。ディスクセット内のすべてのディスクが引き継がれます。予約を実行しているホストで、状態データベースが読み取られ、ディスクセットに構成されている共有ボリュームがアクセス可能になります。この時点で、他のホストがディスクセットを予約していた場合、予約を失うため、パニックが発生することがあります。

通常、ディスクセット内の 2 台のホストは、相互に協力して、ディスクセット内のディスクが一度に 1 台だけのホストによって予約されるようにしています。通常の状況とは、両方のホストが稼働し、相互に通信している状態として定義されます。

注 - ディスクが予期せずに予約されないと判断された（おそらくディスクセットを使用している別のホストが強制的にディスクを取得したため）場合、ホストはパニックを引き起こします。この動作により、2 台のホストが同時に同じディスクにアクセスしようとした場合に発生するようなデータの損失を最小にとどめることができます。

ディスクセットの取得または予約の詳細については、[226 ページの「ディスクセットを取得する方法」](#)を参照してください。

ディスクセットの解放

ディスクセットの解放は、ディスクセット内の物理ディスクの保守を行なう場合に役立つことがあります。ディスクセットが解放されると、ホストからアクセスできなくなります。ディスクセット内の両方のホストがセットを解放した場合、ディスクセット内のどちらのホストもディスクセット内のディスクにアクセスできなくなります。

ディスクセットの解放の詳細については、[228 ページの「ディスクセットを解放する方法」](#)を参照してください。

ディスクセットのインポート

Solaris 9/9/04 リリース以降、`metainport` コマンドを使用して、複製されたディスクセットを含むディスクセットを、ディスクセットでデバイス ID をサポートしている

既存の Solaris Volume Manager 構成にインポートできます。また、`metaimport` コマンドを使用して、インポートに使用可能なディスクセットに関して報告することもできます。

リモートレプリケーションソフトウェアを使用して、複製されたディスクセットが作成されます。`metaimport` コマンドを使用して、複製されたディスクセットがインポートされるようにするには、ディスクセット内の各ディスクの状態データベースの複製を格納するスライスも、複製されたディスクセットの同じスライスに複製される必要があります。これは EFI 以外のディスクの場合はスライス 7、EFI ディスクの場合はスライス 6 に対応します。ディスクセットを複製する前に、複製されるデータのディスク構成と、リモートサイトのディスク構成を一致させます。この手順により、状態データベースの複製とデータの両方が正確に複製されます。

`metaimport` コマンドは、ディスクにボリュームまたは状態データベースの複製が格納されていない場合、ディスクセットにディスクをインポートしません。この状況は、ボリュームまたは状態データベースの複製がディスクに追加されていないか、ディスクから削除されている場合に発生します。この場合、ディスクセットを別のシステムにインポートすると、ディスクがディスクセットから失われていることがわかります。たとえば、Solaris Volume Manager ディスクセットあたり、最大 50 の状態データベースの複製が許可されるとします。ディスクセットに 60 個のディスクがある場合、状態データベースの複製を格納していない 10 個のディスクは、ディスクセットでインポートされるようにするために、ボリュームを格納している必要があります。

ディスクセットのインポートに関するタスクについては、[231 ページの「ディスクセットのインポート」](#)を参照してください。

自動ディスクパーティション分割

ディスクセットに新しいディスクを追加すると、Solaris Volume Manager によって、ディスク形式がチェックされます。必要に応じて、Solaris Volume Manager は、状態データベースの複製用の十分な領域があり、適切に構成されているスライス 7 が含まれるように、ディスクを再分割します。スライス 7 の正確なサイズはディスクのジオメトリによって異なります。ただし、サイズは 4M バイト以上で、6M バイト近くになる可能性があります (シリンダの境界がある場所によりま

す)。デフォルトで、Solaris Volume Manager は状態データベースの複製をスライス 7 に配置します。スライスに複数の状態データベースの複製を収めるために、スライス 7 のデフォルトのサイズを増加するか、状態データベースの複製のサイズを減らすことができます。

注- 状態データベースの複製のサイズや状態データベースの複製に格納される情報のサイズなどのさまざまな要因に基づいて、将来スライス7の最小サイズは変更される可能性があります。複数所有者ディスクセット内の状態データベースの複製のデフォルトのサイズは16Mバイトです。

ディスクセット内で使用するディスクは、以下の条件を満たすスライス7が必要です。

- セクター0から始まる
- ディスクラベルと状態データベースの複製用の十分な領域がある
- マウントできない
- スライス2を含む他のスライスと重複していない

既存のパーティションテーブルがこの条件を満たさない場合、Solaris Volume Managerによってディスクが再分割されます。Solaris Volume Managerが使用するためにスライス7の各ドライブの小さな部分が予約されます。各ドライブの残りの領域はスライス0に配置されます。ディスク上の既存のデータは、再分割によって失われます。

ヒント- ドライブをディスクセットに追加した後に、スライス7がまったく変更されない場合を除いて、必要に応じて再分割できます。

次のprtvtoc コマンドの出力に、ディスクセットに追加される前のディスクを示します。

```
[root@lexicon:apps]$ prtvtoc /dev/rdisk/clt6d0s0
* /dev/rdisk/clt6d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   133 sectors/track
*   27 tracks/cylinder
*   3591 sectors/cylinder
*   4926 cylinders
*   4924 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Partition Tag  Flags  First Sector  Sector Count  Last Sector  Mount Directory
*   0         2     00         0     4111695     4111694
*   1         3     01     4111695     1235304     5346998
*   2         5     01         0     17682084    17682083
*   3         0     00     5346999     4197879     9544877
*   4         0     00     9544878     4197879     13742756
*   5         0     00    13742757     3939327     17682083
```

上の出力は、ディスクにスライス7が含まれていないことを示しています。そのため、ディスクがディスクセットに追加されると、Solaris Volume Manager はディスクを再分割します。次の prtvtoc コマンドの出力に、ディスクセットに追加された後のディスクを示します。

```
[root@lexicon:apps]$ prtvtoc /dev/rdsk/clt6d0s0
* /dev/rdsk/clt6d0s0 partition map
*
* Dimensions:
*   512 bytes/sector
*   133 sectors/track
*   27 tracks/cylinder
*   3591 sectors/cylinder
*   4926 cylinders
*   4924 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*
* Partition Tag  Flags    First   Sector   Last
* Partition Tag  Flags    Sector  Count    Sector  Mount Directory
*   0      0    00    10773 17671311 17682083
*   7      0    01         0    10773    10772
```

出力には、シリンダ0から始まり、状態データベースの複製用の十分な領域があるスライス7を含めるように、ディスクが再分割されたことを示しています。ディスクセットに追加したディスクに、それぞれ受け入れ可能なスライス7がある場合、それらは再分割されません。

注 - Solstice DiskSuite ソフトウェアからアップグレードしたディスクセットがある場合、それらのセット上のデフォルトの状態データベースの複製のサイズは Solaris Volume Manager の 8192 ブロックではなく、1034 ブロックになります。さらに、Solstice DiskSuite ソフトウェアによって追加されたディスク上のスライス7は、それに応じて、Solaris Volume Manager によって追加されたディスク上のスライス7より小さくなります。

ディスクセット名の要件

ディスクセットボリューム名は他の Solaris Volume Manager コンポーネント名と似ています。ただし、ディスクセット名は名前の一部として含まれます。たとえば、ボリュームパス名には、/dev/md/ の後とパス内の実際のボリューム名の前に、ディスクセット名が含まれます。

次の表に、ディスクセットボリューム名の例をいくつか示します。

表 18-1 ディスクセットのボリューム名の例

<code>/dev/md/blue/dsk/d0</code>	ディスクセット blue 内のブロックボリューム d0
<code>/dev/md/blue/dsk/d1</code>	ディスクセット blue 内のブロックボリューム d1
<code>/dev/md/blue/rdsk/d126</code>	ディスクセット blue 内の raw ボリューム d126
<code>/dev/md/blue/rdsk/d127</code>	ディスクセット blue 内の raw ボリューム d127

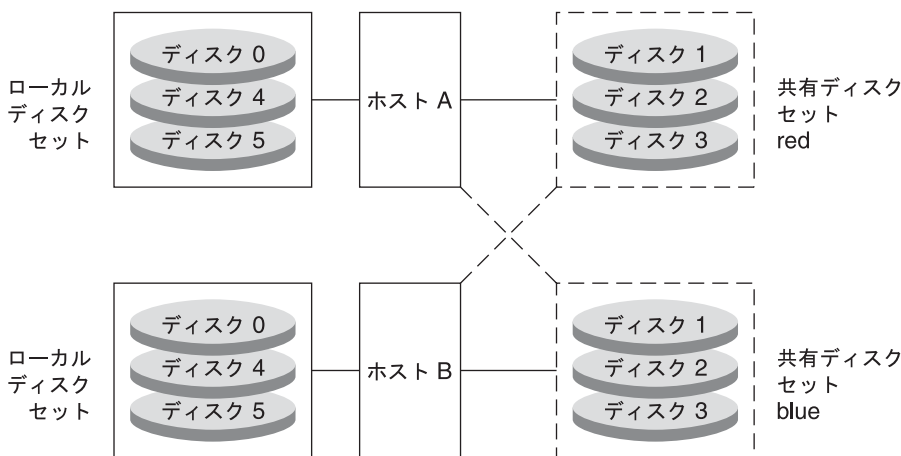
同様に、ホットスペアプールはホットスペア名の一部としてディスクセット名が含まれます。

例 — 2つの共有ディスクセット

図 18-1 に 2つのディスクセットを使用する構成の例を示します。

この構成で、ホスト A とホスト B はディスクセット red と blue を共有しています。それらはそれぞれ、共有されていない独自のローカルディスクセットを使用しています。ホスト A で障害が発生すると、ホスト B はホスト A の共有ディスクセット、ディスクセット red の制御を引き継ぐことができます。同様に、ホスト B で障害が発生すると、ホスト A はホスト B の共有ディスクセット、ディスクセット blue の制御を引き継ぐことができます。

図 18-1 ディスクセットの例



ディスクセットの操作のガイドライン

ディスクセットを操作する場合、次のガイドラインを考慮してください。

- ディスクセットに接続される各ホストで Solaris Volume Manager を構成する必要があります。
- 各ホストでは、ディスクセットを作成する前に、ローカル状態データベースを設定しておく必要があります。
- ディスクセットを作成し、そのディスクセットのボリュームを作成する手順の順番は、まず、ディスクセットを作成することです。次に、ディスクセットにディスクを追加します。最後に、ディスクセットにボリュームを作成します。
- クラスタ環境でディスクセットを作成し、操作するには、root がグループ 14 のメンバーである必要があります。または、各ホスト上の `/rhosts` ファイルに、ディスクセットに関連付けられている他のホスト名のエントリが含まれている必要があります。

注 - SunCluster 3.x 環境では、この手順は必要ありません。

- ディスクセットの保守を実行するには、ホストがディスクセットの所有者であるか、ディスクセットを予約している必要があります。ホストがディスクセットの暗黙的所有権を取得するには、最初のドライブをディスクセットに入れます。
- ファイルシステム、データベース、またはその他のアプリケーションで使用しているドライブをディスクセットに追加することはできません。ドライブを追加する前に、それが現在使用されていないことを確認します。
- 保持したい既存のデータを格納しているドライブをディスクセットに追加しないでください。ディスクをディスクセットに追加するプロセスによって、ディスクが再分割され、既存のデータが破棄されます。
- ローカルボリューム管理と異なり、ディスクセットに状態データベースの複製を手動で作成したり、削除したりする必要はありません。Solaris Volume Manager は、ディスクセット内のすべてのドライブで、状態データベースの複製の適切な数のバランスを取ろうとします。
- ディスクセットにドライブが追加されると、Solaris Volume Manager は、残りのドライブで、状態データベースの複製のバランスを取り直します。後で、必要に応じて、`metadb` コマンドを使用して、複製のレイアウトを変更できます。

ディスクセット内の非同期共有ストレージ

Solaris Volume Manager の以前のバージョンでは、ディスクセット内のホスト間で共有する予定のあるすべてのディスクを各ホストに接続する必要がありました。さらに、それらは各ホストで、正確に同じパス、ドライバ、名前を持つ必要がありました。特に、共有ディスクドライブは、両方のホストから、同じ場所 (/dev/rdsk/c#t#d#) で参照される必要がありました。また、共有ディスクは同じドライバ名 (ssd) を使用する必要がありました。

現在の Solaris OS リリースでは、一般にアクセス可能なストレージのさまざまなビューを持つシステムで、ディスクセットへのアクセスを非同期で共有できます。ディスクセットのデバイス ID のサポートの導入により、Solaris Volume Manager は名前付きディスクセット内のディスクの移動を自動的に追跡します。

最新の Solaris OS にアップグレードする場合、ディスクの追跡を有効にするために、ディスクセットを 1 回取得する必要があります。ディスクセットの取得の詳細については、[226 ページの「ディスクセットを取得する方法」](#)を参照してください。

auto-take 機能が有効にされていない場合、各ディスクセットを手動で取得する必要があります。この機能が有効にされている場合、この手順は、システムのリブート時に自動的に行なわれます。auto-take 機能の詳細については、[207 ページの「auto-take ディスクセット」](#)を参照してください。

さらに、この拡張されたデバイス ID のサポートにより、別のシステム上に作成されたディスクセットでもインポートできます。ディスクセットのインポートの詳細については、[210 ページの「ディスクセットのインポート」](#)を参照してください。

シナリオ-ディスクセット

次の例では、[第 5 章「Solaris Volume Manager の構成と使用 \(シナリオ\)」](#)に示すサンプルシステムを利用して、SAN (Storage Area Network) ファブリック上に存在するストレージを管理するために、どのようにディスクセットを使用すべきかを示します。

サンプルシステムには、ファイバスイッチと SAN ストレージに接続されている追加のコントローラーがあるとします。SAN ファブリック上のストレージは、SCSI および IDE ディスクなどの他のデバイスのようにブートプロセスの早期に、システムから使用できません。さらに、Solaris Volume Manager はブート時に、ファブリック上の論理ボリュームが利用不可であると報告します。ただし、ストレージをディスクセットに追加し、ディスクセットツールを使用してストレージを管理することによって、ブート時の使用可能性に関するこの問題が回避されます。さらに、ファブリック接続されたストレージは、ローカルストレージから、個別のディスクセットによって制御される名前空間内で簡単に管理できます。

ディスクセット (タスク)

この章では、ディスクセットに関連したタスクの実行について説明します。これらのタスクに関する概念については、第 18 章「ディスクセット (概要)」を参照してください。

ディスクセット (タスクマップ)

次のタスクマップに、Solaris Volume Manager のディスクセットおよび Solaris Volume Manager for Sun Cluster の複数所有者ディスクセットを管理するために必要な手順を示します。注記がある場合を除き、すべてのコマンドが両方のタイプのディスクセットに対して機能します。Solaris Volume Manager GUI は、複数所有者ディスクセットに関連したタスクには使用できません。

タスク	説明	参照先
ディスクセットを作成する	ディスクセットを作成するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。 複数所有者ディスクセットを作成するには、 <code>metaset -M</code> コマンドを使用します。	219 ページの「ディスクセットを作成する方法」
ディスクセットにディスクを追加する	ディスクセットにディスクを追加するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	220 ページの「ディスクセットにディスクを追加する方法」
ディスクセットにホストを追加する	ディスクセットにホストを追加するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	222 ページの「ディスクセットに別のホストを追加する方法」

タスク	説明	参照先
ディスクセット内に Solaris Volume Manager ボリュームを作成する	ディスクセット内にボリュームを作成するには、Solaris Volume Manager GUI または <code>metainit</code> コマンドを使用します。	223 ページの「ディスクセット内に Solaris Volume Manager コンポーネントを作成する方法」
ディスクセットのステータスを確認する	ディスクセットのステータスを確認するには、Solaris Volume Manager GUI を使用するか、または <code>metaset</code> コマンドを使用します。	224 ページの「ディスクセットのステータスを確認する方法」
ディスクセットからディスクを削除する	ディスクセットからディスクを削除するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	225 ページの「ディスクセットからディスクを削除する方法」
ディスクセットを取得する	ディスクセットを取得するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	226 ページの「ディスクセットを取得する方法」
ディスクセットを解放する	ディスクセットを解放するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	228 ページの「ディスクセットを解放する方法」
ディスクセットからホストを削除する	ディスクセットからホストを削除するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	229 ページの「ホストまたはディスクセットを削除する方法」
ディスクセットを削除する	ディスクセットから最後のホストを削除し、それによってディスクセットを削除するには、Solaris Volume Manager GUI または <code>metaset</code> コマンドを使用します。	229 ページの「ホストまたはディスクセットを削除する方法」
ディスクセットをインポートする	ディスクセットに関するレポートを実行してどのディスクセットをインポートできるかを判定したり、ディスクセットをあるシステムから別のシステムにインポートしたりするには、 <code>metainport</code> コマンドを使用します。	231 ページの「ディスクセットのインポート」

ディスクセットの作成

▼ ディスクセットを作成する方法

始める前に [215 ページの「ディスクセットの操作のガイドライン」](#)を確認してください。

- 1 ディスクセットを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。「アクション」、「ディスクセットを作成」の順に選択します。次に、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
 - コマンド行からディスクセットを最初から作成するには、次の形式の `metaset` コマンドを使用します。

```
# metaset -s diskset-name -a -h -M hostname
```

<code>-s diskset-name</code>	<code>metaset</code> コマンドの対象となるディスクセットの名前を指定します。
<code>-a</code>	指定されたディスクセットにホストを追加します。Solaris Volume Manager は、ディスクセットあたり最大 4 つのホストをサポートしています。
<code>-M</code>	作成されるディスクセットが複数所有者ディスクセットであることを指定します。
<code>-h hostname</code>	ディスクセットに追加される 1 つまたは複数のホストを指定します。最初のホストを追加すると、ディスクセットが作成されます。あとで、2 番目のホストを追加できます。ただし、指定された <code>hostname</code> 上にこのディスクセット内のすべてのディスクが見つからない場合、2 番目のホストは受け入れられません。 <code>hostname</code> は、 <code>/etc/nodename</code> ファイル内にあるのと同じ名前です。

詳細は、[metaset\(1M\)](#) のマニュアルページを参照してください。

- 2 新しいディスクセットのステータスを確認します。

```
# metaset
```

例 19-1 ディスクセットの作成

次の例では、ホスト `host1` から、`blue` という名前の共有ディスクセットを作成します。`metaset` コマンドは、ステータスを表示します。この時点では、ディスクセットに所有者はいません。ディスクセットにディスクを追加したホストが、デフォルトで所有者になります。

```
# metaset -s blue -a -h host1
# metaset
Set name = blue, Set number = 1

Host                Owner
  host1
```

例 19-2 複数所有者ディスクセットの作成

次の例では、`red` という名前の複数所有者ディスクセットを作成します。`metaset` コマンドからの出力の 1 行目には、このディスクセットが複数所有者ディスクセットであることを示す「Multi-owner」が表示されています。

```
# metaset -s red -a -M -h nodeone
# metaset -s red
Multi-owner Set name = red, Set number = 1, Master =

Host                Owner                Member
  nodeone
```

ディスクセットの拡張

▼ ディスクセットにディスクを追加する方法



注意 - 32ビットカーネルで Solaris ソフトウェアを実行することを予定している場合や、Solaris 9 4/03 リリースより前のバージョンの Solaris OS の使用を予定している場合は、ディスクセットに 1T バイトを超えるディスクを追加しないでください。Solaris Volume Manager でのマルチテラバイトボリュームのサポートの詳細は、[52 ページの「Solaris Volume Manager でのマルチテラバイトのサポートの概要」](#)を参照してください。

ディスクセットに追加できるのは、次の条件を満たすディスクだけです。

- ディスクがボリュームまたはホットスペアプールで使用されていない。

- ディスクに状態データベースの複製が含まれていてはいけません。
- ディスクが現在マウントされていたり、スワップに使用されていたり、アプリケーションで使用するためにそれ以外の方法で開かれていたりしてはいけません。

始める前に 215 ページの「ディスクセットの操作のガイドライン」を確認してください。

- 1 ディスクセットにディスクを追加するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。変更するディスクセットを選択します。次に、マウスの右ボタンをクリックして「プロパティ」を選択します。「ディスク」タブを選択します。「ディスクを追加」をクリックします。次に、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
 - コマンド行からディスクセットにディスクを追加するには、次の形式の `metaset` コマンドを使用します。

```
# metaset -s diskset-name -a disk-name
```

`-s diskset-name` metaset コマンドの対象となるディスクセットの名前を指定します。

`-a` 指定されたディスクセットにディスクを追加します。

`disk-name` ディスクセットに追加するディスクを指定します。ディスク名の形式は、`cxtxdx` です。ディスクセットにディスクを追加する場合、「`sx`」のスライス識別子は含まれません。

詳細は、[metaset\(1M\)](#) のマニュアルページを参照してください。

ディスクセットにディスクを追加した最初のホストが、そのディスクセットの所有者になります。



注意-データを含むディスクをディスクセットに追加しないでください。データを含むディスクをディスクセットに追加すると、ディスクが再パーティション分割され、データが破棄されることがあります。

- 2 ディスクセットとディスクのステータスを確認します。

```
# metaset
```

例 19-3 ディスクセットへのディスクの追加

```
# metaset -s blue -a c1t6d0
# metaset
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
Drive	Dbase
c1t6d0	Yes

この例では、ホスト名はhost1です。共有ディスクセットはblueです。ディスクc1t6d0のみがディスクセットblueに追加されました。

必要に応じて、コマンド行に各ディスクのリストを指定することによって、一度に複数のディスクを追加できます。たとえば、次のコマンドを使用すると、ディスクセットに同時に2つのディスクを追加できます。

```
# metaset -s blue -a c1t6d0 c2t6d0
```

▼ ディスクセットに別のホストを追加する方法

既存のディスクセットに別のホストを追加する方法について説明します。Solaris Volume Managerは、ディスクセットあたり最大4つのホストをサポートしています。

始める前に [215 ページの「ディスクセットの操作のガイドライン」](#)を確認してください。

1 ディスクセットにホストを追加するには、次のいずれかの方法を使用します。

- Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開き、変更するディスクセットを選択します。変更するディスクセットを選択します。次に、マウスの右ボタンをクリックして「プロパティ」を選択します。「ホスト」タブを選択します。「ホストの追加」をクリックします。次に、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
- コマンド行からディスクセットにホストを追加するには、次の形式のmetaset コマンドを使用します。

```
# metaset -s diskset-name -a -h hostname
```

-s *diskset-name* ホストを追加するディスクセットの名前を指定します。

-a 指定されたディスクセットにホストを追加します。

-h *hostname* ディスクセットに追加される1つまたは複数のホスト名を指定します。最初のホストを追加すると、ディスクセットが作成されます。このホスト名は、`/etc/nodename` ファイル内にあるのと同じ名前です。

詳細は、[metaset\(1M\)](#)のマニュアルページを参照してください。

- 2 ホストがディスクセットに追加されていることを確認します。

```
# metaset
```

例 19-4 ディスクセットへの別のホストの追加

```
# metaset -s blue -a -h host2
# metaset
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
host2	

Drive	Dbase
c1t6d0	Yes
c2t6d0	Yes

この例は、ディスクセット blue へのホスト host2 の追加を示しています。

▼ ディスクセット内に **Solaris Volume Manager** コンポーネントを作成する方法

ディスクセットを作成したあと、そのディスクセットに追加したディスクを使用してボリュームとホットスペアプールを作成できます。Solaris 管理コンソール内の拡張ストレージツールまたはコマンド行ユーティリティのどちらかを使用できます。

- ディスクセット内にボリュームやその他の **Solaris Volume Manager** コンポーネントを作成するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」、「状態データベース複製」、または「ホットスペアプール」ノードを開きます。「アクション」、「作成」の順に選択します。次に、ウィザードの指示に従います。詳細は、オンラインヘルプを参照してください。
 - 同じ基本的な構文を持つ同じコマンド行ユーティリティを使用して、ボリューム、状態データベースの複製、またはホットスペアプールを作成します。ただし、すべてのコマンドについて、コマンドの直後に `-s disk-set` を追加します。

```
# command -s disk-set
```

例 19-5 ディスクセット内での Solaris Volume Manager ボリュームの作成

次の例は、ディスクセット blue 内でのミラー d10 の作成を示しています。このミラーは、サブミラー (RAID-0 ボリューム) d11 と d12 で構成されています。

```
# metainit -s blue d11 1 1 c1t6d0s0
blue/d11: Concat/Stripe is setup
# metainit -s blue d12 1 1 c2t6d0s0
blue/d12: Concat/Stripe is setup
# metainit -s blue d10 -m d11
blue/d10: Mirror is setup
# metattach -s blue d10 d12
blue/d10: submirror blue/d12 is attached

# metastat -s blue
blue/d10: Mirror
  Submirror 0: blue/d11
    State: Okay
  Submirror 1: blue/d12
    State: Resyncing
  Resync in progress: 0 % done
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 17674902 blocks

blue/d11: Submirror of blue/d10
  State: Okay
  Size: 17674902 blocks
  Stripe 0:
    Device                Start Block  Dbase State      Reloc Hot Spare
    c1t6d0s0                0           No  Okay

blue/d12: Submirror of blue/d10
  State: Resyncing
  Size: 17674902 blocks
  Stripe 0:
    Device                Start Block  Dbase State      Reloc Hot Spare
    c2t6d0s0                0           No  Okay
```

ディスクセットの保守

▼ ディスクセットのステータスを確認する方法

- ディスクセットのステータスを確認するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。マウスの右ボタンで、モニターするディスクセットをクリックします。次に、メニューから「プロパティ」を選択します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metaset` コマンドを使用して、ディスクセットステータスを表示します。

```
# metaset -s diskset-name
```


詳細は、[metaset\(1M\)](#)のマニュアルページを参照してください。

注-ディスクセットの所有権は、所有しているホストでのみ表示されます。

例 19-6 指定されたディスクセットのステータスの確認

次の例は、`-s` オプションに続けてディスクセット `blue` の名前が指定された `metaset` コマンドを示しています。このコマンドからの出力には、そのディスクセットのステータス情報が表示されます。この出力は、`host1` がディスクセットの所有者であることを示しています。`metaset` コマンドではまた、ディスクセット内のディスクも表示されます。

```
red# metaset -s blue

Set name = blue, Set number = 1

Host                Owner
  host1              Yes

Drive               Dbase
  c1t6d0             Yes
  c2t6d0             Yes
```

`metaset` コマンドは、それ自体で、すべてのディスクセットのステータスを表示します。

▼ ディスクセットからディスクを削除する方法

- 1 ディスクセットからディスクを削除するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。マウスの右ボタンで、解放するディスクセットをクリックします。次に、メニューから「プロパティ」を選択します。「ディスク」タブをクリックします。オンラインヘルプの指示に従います。
 - 次の形式の `metaset` コマンドを使用して、ディスクセットからディスクを削除します。

```
# metaset -s diskset-name -d disk-name
-s diskset-name   ディスクを削除するディスクセットの名前を指定します。
-d disk-name      ディスクセットから削除するディスクを指定します。ディスク
                  名の形式は、cxtxdx です。ディスクセットからディスクを削除
                  する場合、「sx」のスライス識別子は含まれません。
```

詳細は、[metaset\(1M\)](#)のマニュアルページを参照してください。

- 2 ディスクがディスクセットから削除されていることを確認します。

```
# metaset -s diskset-name
```

注- ディスクセットを削除するには、まずそのディスクセットからすべてのディスクを削除する必要があります。

例 19-7 ディスクセットからのディスクの削除

次の例は、ディスクセット blue からディスク c1t6d0 の削除を示しています。

```
host1# metaset -s blue -d c1t6d0
host1# metaset -s blue

Set name = blue, Set number = 1

Host                Owner
  host1
  host2

Drive               Dbase
  c2t6d0             Yes
```

▼ ディスクセットを取得する方法

注- このオプションは、複数所有者ディスクセットには使用できません。

- ディスクセットを取得するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。マウスの右ボタンで、取得するディスクセットをクリックします。次に、メニューから「所有権を取得」を選択します。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metaset` コマンドを使用します。

```
# metaset -s diskset-name -t -f

-s diskset-name   取得するディスクセットの名前を指定します。
-t               ディスクセットを取得するように指定します。
-f               ディスクセットを強制的に取得するように指定します。
```

詳細は、[metaset\(1M\)](#)のマニュアルページを参照してください。

一度に1つのホストだけがディスクセットを所有できます。ディスクセット内の1つのホストがそのディスクセットを取得すると、ディスクセット内のその他のホストは、そのディスクセット内のディスク上のデータにアクセスできません。

`metaset` コマンドのデフォルトの動作では、ディスクセットの所有権を持つホスト上で解放が可能である場合にのみ、ほかのホストがディスクセットを取得できるようになります。ディスクセットを強制的に取得するには、`-f` オプションを使用します。このオプションは、別のホストが現在ディスクセットを所有しているかどうかにかかわらず、ディスクセットを取得します。この方法は、ディスクセット内のホストがダウンしているか、または通信していない場合に使用します。この時点でほかのホストがディスクセットを取得された場合、そのホストは、そのディスクセットに対する入出力操作を実行しようとするするとパニックを起こします。

注-ディスクセットの所有権は、所有しているホストでのみ表示されます。

例 19-8 ディスクセットの取得

次の例では、ホスト `host1` がホスト `host2` と通信します。この通信によって、ホスト `host1` がディスクセットを取得しようとする前に、ホスト `host2` がそのディスクセットを解放していることが保証されます。

```
host1# metaset
...
Set name = blue, Set number = 1

Host                Owner
  host1
  host2
...
host1# metaset -s blue -t
host1# metaset
...
Set name = blue, Set number = 1

Host                Owner
  host1                Yes
  host2
...

```

`host2` がディスクセット `blue` を所有していたとすると、前の出力の「Owner」列は引き続き空白になります。`metaset` コマンドは、このコマンドを発行しているホストがそのディスクセットを所有しているかどうかのみを示します。

例 19-9 ディスクセットの強制的な取得

次の例では、ディスクセットを取得しているホストがその他のホストと通信しません。代わりに、`-f` オプションにより、ディスクセット内のディスクを警告なしで強

制的に取得できます。ほかのホストがそのディスクセットを所有していた場合、そのホストは、そのディスクセットに対する入出力操作を試行したとするとパニックを起こします。

```
# metaset -s blue -t -f
```

▼ ディスクセットを解放する方法

ディスクセットの解放は、ディスクセット内の物理ディスクに対する保守を実行する場合に役立ちます。ディスクセットが解放されると、ホストからアクセスできなくなります。ディスクセット内の両方のホストがそのディスクセットを解放した場合は、ディスクセット内のどちらのホストも、そのディスクセットで定義されているボリュームまたはホットスペアプールに直接にはアクセスできません。ただし、両方のホストがディスクセットを解放した場合、各ホストは自身の `c*t*d*` 名を使用してそれらのディスクに直接アクセスできます。

注- このオプションは、複数所有者ディスクセットには使用できません。

始める前に [215 ページの「ディスクセットの操作のガイドライン」](#)を確認してください。

- 1 ディスクセットを解放するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。マウスの右ボタンで、解放するディスクセットをクリックします。次に、メニューから「所有権を解放」を選択します。詳細は、オンラインヘルプを参照してください。
 - ディスクセットの所有権を解放するには、次の形式の `metaset` コマンドを使用します。

```
# metaset -s diskset-name -r
```

`-s diskset-name` `metaset` コマンドの対象となるディスクセットの名前を指定します。

`-r` ディスクセットの所有権を解放します。ディスクセット内のすべてのディスクの予約が削除されます。そのディスクセット内のボリュームにはアクセスできなくなります。

詳細は、[metaset\(1M\)](#)のマニュアルページを参照してください。

注- ディスクセットの所有権は、所有しているホストでのみ表示されます。

- このホスト上でディスクセットが解放されていることを確認します。

```
# metaset
```

例 19-10 ディスクセットの解放

次の例は、ディスクセット `blue` の解放を示しています。ディスクセットの所有者が存在しないことに注意してください。ホスト `host1` からのステータスの表示は誤解を招く可能性があります。ホストが判定できるのは、自身がディスクセットを所有しているかどうかだけです。たとえば、ホスト `host2` がディスクセットの所有権を取得しても、ホスト `host1` からはその所有権が表示されません。 `host2` がディスクセットの所有権を持っていることを表示するのは、ホスト `host2` だけです。

```
host1# metaset -s blue -r
host1# metaset -s blue

Set name = blue, Set number = 1

Host                Owner
  host1
  host2

Drive              Dbase
  clt6d0            Yes
  c2t6d0            Yes
```

▼ ホストまたはディスクセットを削除する方法

ディスクセットを削除するには、そのディスクセットにディスクが含まれていないこと、およびそのディスクセットにほかのホストが接続されていないことが必要です。最後のホストを削除すると、そのディスクセットが破棄されます。

- ディスクセットからホストを削除するか、またはディスクセットを削除するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ディスクセット」ノードを開きます。マウスの右ボタンで、解放するディスクセットをクリックし、メニューから「削除」を選択します。オンラインヘルプの指示に従います。
 - ホストを削除するには、次の形式の `metaset` コマンドを使用します。

```
metaset -s diskset-name -d -h hostname
-s diskset-name  metaset コマンドの対象となるディスクセットの名前を指定し
                  ます。
-d              ディスクセットからホストを削除します。
```

`-h hostname` 削除するホストの名前を指定します。

ディスクセットを削除するには、前の `metaset` コマンドと同じ形式を使用します。ディスクセットを削除するには、そのディスクセットにディスクが含まれていないこと、およびほかのホストがそのディスクセットを所有していないことが必要です。最後のホストを削除すると、そのディスクセットが破棄されます。

詳細は、[metaset\(1M\)](#) のマニュアルページを参照してください。

- 2 `metaset` コマンドを使用して、ホストがディスクセットから削除されていることを確認します。現在の(所有している)ホストだけが表示されることに注意してください。その他のホストは削除されています。

```
# metaset -s disk-set
```

例 19-11 ディスクセットからのホストの削除

次の例は、ディスクセット `blue` からのホスト `host2` の削除を示しています。

```
# metaset -s blue
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes
..host2	

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t4d0	Yes
c1t5d0	Yes
c1t6d0	Yes
c2t1d0	Yes

```
# metaset -s blue -d -h host2
```

```
# metaset -s blue
Set name = blue, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t4d0	Yes
c1t5d0	Yes
c1t6d0	Yes
c2t1d0	Yes

例 19-12 ディスクセットからの最後のホストの削除

次の例は、ディスクセット blue からの最後のホストの削除を示しています。

```
host1# metaset -s blue -d -h host1
host1# metaset -s blue

metaset: host: setname "blue": no such set
```

ディスクセットのインポート

ディスクセットのインポート

metaimport コマンドを使用すると、ディスクセットをあるシステムから別のシステムにインポートできます。

▼ インポートに使用可能なディスクセットに関するレポートを出力する方法

- 1 スーパーユーザーになります。
- 2 インポートに使用可能なディスクセットに関するレポートを取得します。

```
# metaimport -r -v
-r システムへのインポートに使用可能な構成解除されたディスクセットのレポートを提供します。
-v 状態データベース (metadb) の複製の場所や、システムへのインポートに使用可能な構成解除されたディスクセットのディスクのステータスに関する詳細情報を提供します。
```

例 19-13 インポートに使用可能なディスクセットに関する報告

次の例は、インポートに使用可能なディスクセットに関するレポートを出力する方法を示しています。metaimport コマンドからの出力では、通常のディスクセットと複製されたディスクセットが区別されます。

```
# metaimport -r
# metaimport -r
Drives in regular diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
More info:
  metaimport -r -v c1t2d0
```

```

Import:  metainport -s <newsetname> c1t2d0
        Drives in replicated diskset including disk c1t4d0:
            c1t4d0
            c1t5d0
        More info:
            metainport -r -v c1t4d0
Import:  metainport -s <newsetname> c1t4d0
# metainport -r -v c1t2d0
Import:  metainport -s <newsetname> c1t2d0
Last update: Mon Dec 29 14:13:35 2003
Device      offset      length replica flags
c1t2d0       16          8192    a      u
c1t3d0       16          8192    a      u

```

▼ ディスクセットをあるシステムから別のシステムにインポートする方法

- 1 スーパーユーザーになります。
- 2 ディスクセットがインポートに使用可能であることを確認します。
metainport -r -v
- 3 使用可能なディスクセットをインポートします。
metainport -s diskset-name disk-name
-s *diskset-name* 作成されるディスクセットの名前を指定します。
disk-name インポートされるディスクセットの状態データベースの複製が含まれているディスク (c#t#d#) を指定します。
- 4 ディスクセットがインポートされていることを確認します。
metaset -s diskset-name

例 19-14 ディスクセットのインポート

次の例は、ディスクセットをインポートする方法を示しています。

```

# metainport -s red c1t2d0
Drives in diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
  c1t8d0
More info:
  metainport -r -v c1t2d0
# metaset -s red

```

Set name = red, Set number = 1

Host	Owner
host1	Yes
Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t8d0	Yes

Solaris Volume Manager の保守 (タスク)

この章では、Solaris Volume Manager に関する一般的なストレージ管理保守タスクの実行について説明します。

この章の内容は次のとおりです。

- 235 ページの「Solaris Volume Manager の保守 (タスクマップ)」
- 236 ページの「Solaris Volume Manager 構成の表示」
- 240 ページの「ボリュームの名前の変更」
- 243 ページの「構成ファイルの操作」
- 245 ページの「Solaris Volume Manager のデフォルト値の変更」
- 246 ページの「growfs コマンドを使用したファイルシステムの拡張」
- 248 ページの「RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要」

Solaris Volume Manager の保守 (タスクマップ)

次のタスクマップに、Solaris Volume Manager を保守するために必要な手順を示します。

タスク	説明	参照先
Solaris Volume Manager 構成を表示する	システム構成を表示するには、Solaris Volume Manager GUI または <code>metastat</code> コマンドを使用します。	236 ページの「Solaris Volume Manager ボリューム構成を表示する方法」
ボリュームの名前を変更する	ボリュームの名前を変更するには、Solaris Volume Manager GUI または <code>metarename</code> コマンドを使用します。	241 ページの「ボリュームの名前を変更する方法」
構成ファイルを作成する	構成ファイルを作成するには、 <code>metastat -p</code> コマンドおよび <code>metadb</code> コマンドを使用します。	243 ページの「構成ファイルを作成する方法」

タスク	説明	参照先
構成ファイルから Solaris Volume Manager を初期化する	構成ファイルから Solaris Volume Manager を初期化するには、metainit コマンドを使用します。	243 ページの「構成ファイルから Solaris Volume Manager を初期化する方法」
ファイルシステムを拡張する	ファイルシステムを拡張するには、growfs コマンドを使用します。	247 ページの「ファイルシステムを拡張する方法」
コンポーネントを有効にする	コンポーネントを有効にするには、Solaris Volume Manager GUI または metareplace コマンドを使用します。	249 ページの「コンポーネントの有効化」
コンポーネントを置き換える	コンポーネントを置き換えるには、Solaris Volume Manager GUI または metareplace コマンドを使用します。	249 ページの「コンポーネントを別の使用可能なコンポーネントに置き換える」

Solaris Volume Manager 構成の表示

ヒント-metastat コマンドは、出力をソートしません。より管理しやすい構成のリストを取得するには、metastat -p コマンドの出力を sort または grep コマンドにパイプしてください。

▼ Solaris Volume Manager ボリューム構成を表示する方法

- ボリューム構成を表示するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリューム」ノードを開きます。詳細は、オンラインヘルプを参照してください。
 - 次の形式の metastat コマンドを使用します。

```
# metastat -p -i component-name
```

-p 出力を圧縮されたサマリーで表示するように指定します。この出力は、md.tab ファイルの作成での使用に適しています。

-i RAID-1 (ミラー) ボリューム、RAID-5 ボリューム、およびホットスペアにアクセスできることを確認するように指定します。

component-name 表示するボリュームの名前を指定します。ボリューム名が指定されていない場合は、コンポーネントの完全なリストが表示されます。

例 20-1 Solaris Volume Manager ボリューム構成の表示

次の例は、metastat コマンドからの出力を示しています。

```
# metastat
d50: RAID
  State: Okay
  Interlace: 32 blocks
  Size: 20985804 blocks
Original device:
  Size: 20987680 blocks
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t4d0s5        330         No  Okay       Yes
  c1t5d0s5        330         No  Okay       Yes
  c2t4d0s5        330         No  Okay       Yes
  c2t5d0s5        330         No  Okay       Yes
  c1t1d0s5        330         No  Okay       Yes
  c2t1d0s5        330         No  Okay       Yes

d1: Concat/Stripe
  Size: 4197879 blocks
  Stripe 0:
  Device          Start Block  Dbase  Reloc
  c1t2d0s3        0           No     Yes

d2: Concat/Stripe
  Size: 4197879 blocks
  Stripe 0:
  Device          Start Block  Dbase  Reloc
  c2t2d0s3        0           No     Yes

d80: Soft Partition
  Device: d70
  State: Okay
  Size: 2097152 blocks
  Extent          Start Block          Block count
  0               1                    2097152

d81: Soft Partition
  Device: d70
  State: Okay
  Size: 2097152 blocks
  Extent          Start Block          Block count
  0               2097154              2097152

d70: Mirror
  Submirror 0: d71
  State: Okay
  Submirror 1: d72
  State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 12593637 blocks

d71: Submirror of d70
```

```

State: Okay
Size: 12593637 blocks
Stripe 0:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t3d0s3        0           No  Okay        Yes
Stripe 1:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t3d0s4        0           No  Okay        Yes
Stripe 2:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t3d0s5        0           No  Okay        Yes

d72: Submirror of d70
State: Okay
Size: 12593637 blocks
Stripe 0:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c2t3d0s3        0           No  Okay        Yes
Stripe 1:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c2t3d0s4        0           No  Okay        Yes
Stripe 2:
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c2t3d0s5        0           No  Okay        Yes

hsp010: is empty

hsp014: 2 hot spares
Device          Status      Length      Reloc
c1t2d0s1        Available  617652 blocks Yes
c2t2d0s1        Available  617652 blocks Yes

hsp050: 2 hot spares
Device          Status      Length      Reloc
c1t2d0s5        Available  4197879 blocks Yes
c2t2d0s5        Available  4197879 blocks Yes

hsp070: 2 hot spares
Device          Status      Length      Reloc
c1t2d0s4        Available  4197879 blocks Yes
c2t2d0s4        Available  4197879 blocks Yes

Device Relocation Information:
Device          Reloc      Device ID
c1t2d0          Yes        id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N1S200002103AF29
c2t2d0          Yes        id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0P64Z00002105Q6J7
c1t1d0          Yes        id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N1EM00002104NP2J
c2t1d0          Yes        id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0N93J000071040L3S
c0t0d0          Yes        id1,dad@s53554e575f4154415f5f53543339313430412525415933

```

例 20-2 マルチテラバイト Solaris Volume Manager ボリュームの表示

次の例は、マルチテラバイトストレージボリューム (11T バイト) に対する `metastat` コマンドからの出力を示しています。

```
# metastat d0
d0: Concat/Stripe
  Size: 25074708480 blocks (11 TB)
  Stripe 0: (interlace: 32 blocks)
    Device      Start Block  Dbase  Reloc
    c27t8d3s0    0            No     Yes
    c4t7d0s0     12288        No     Yes
  Stripe 1: (interlace: 32 blocks)
    Device      Start Block  Dbase  Reloc
    c13t2d1s0    16384        No     Yes
    c13t4d1s0    16384        No     Yes
    c13t6d1s0    16384        No     Yes
    c13t8d1s0    16384        No     Yes
    c16t3d0s0    16384        No     Yes
    c16t5d0s0    16384        No     Yes
    c16t7d0s0    16384        No     Yes
    c20t4d1s0    16384        No     Yes
    c20t6d1s0    16384        No     Yes
    c20t8d1s0    16384        No     Yes
    c9t1d0s0     16384        No     Yes
    c9t3d0s0     16384        No     Yes
    c9t5d0s0     16384        No     Yes
    c9t7d0s0     16384        No     Yes
  Stripe 2: (interlace: 32 blocks)
    Device      Start Block  Dbase  Reloc
    c27t8d2s0    16384        No     Yes
    c4t7d1s0     16384        No     Yes
  Stripe 3: (interlace: 32 blocks)
    Device      Start Block  Dbase  Reloc
    c10t7d0s0    32768        No     Yes
    c11t5d0s0    32768        No     Yes
    c12t2d1s0    32768        No     Yes
    c14t1d0s0    32768        No     Yes
    c15t8d1s0    32768        No     Yes
    c17t3d0s0    32768        No     Yes
    c18t6d1s0    32768        No     Yes
    c19t4d1s0    32768        No     Yes
    c1t5d0s0     32768        No     Yes
    c2t6d1s0     32768        No     Yes
    c3t4d1s0     32768        No     Yes
    c5t2d1s0     32768        No     Yes
    c6t1d0s0     32768        No     Yes
    c8t3d0s0     32768        No     Yes
```

次に進む手順

詳細は、[metastat\(1M\)](#) のマニュアルページを参照してください。

ボリュームの名前の変更

ボリュームの名前の変更の背景情報

Solaris Volume Manager では、いくつかの制約に従って、ほとんどのタイプのボリュームの名前をいつでも変更できます。Solaris 管理コンソール内の拡張ストレージツールまたはコマンド行 (`metarename(1M)` コマンド) のどちらかを使用して、ボリュームの名前を変更できます。

ボリュームの名前の変更またはボリューム名の切り替えは、ボリューム名を管理するための管理の利便性のために用意されています。たとえば、すべてのファイルシステムマウントポイントを目的の番号の範囲で整列できます。現在の論理ボリュームの名付けスキームを維持したり、トランザクションボリュームで基となるボリュームの名前と同じ名前を使用できるようにしたりするために、ボリュームの名前を変更する場合があります。

注 - Solaris Volume Manager では、トランザクションボリュームは有効でなくなりしました。それらを置き換えるために、トランザクションボリュームの名前を変更できます。

ボリュームの名前を変更する前に、そのボリュームが現在使用されていないことを確認してください。ファイルシステムの場合は、`swap` としてマウントまたは使用されていないことを確認してください。raw デバイスを使用するその他のアプリケーション (データベースなど) は、データへのアクセスを停止するための独自の方法を備えているべきです。

ボリュームの名前を変更するときの固有の考慮事項には、次のものがあります。

- 次を除く、任意のボリュームの名前を変更できます。
 - ソフトパーティション
 - ソフトパーティションが直接構築されているボリューム
 - ログデバイスとして使用されているボリューム
 - ホットスペアプール
- ディスクセット内のボリュームの名前を変更できます。ただし、ボリュームをあるディスクセットから別のディスクセットに移動するために名前を変更することはできません。

ボリューム名の交換

`metarename` コマンドを `-x` オプションとともに使用すると、親子関係を持つボリュームの名前が交換されます。詳細は、[241 ページの「ボリュームの名前を変更する方法」](#) および [metarename\(1M\)](#) のマニュアルページを参照してください。既存のボリュームの名前が、そのサブコンポーネントの1つと交換されます。たとえば、ミラーとそのいずれかのサブミラーの間でこのタイプの交換が発生する場合があります。`metarename -x` コマンドを使用すると、既存のボリュームのミラー化またはミラー化の解除が容易になります。

注- ボリューム名を交換するには、コマンド行を使用する必要があります。この機能は現在、Solaris Volume Manager GUI では使用できません。ただし、ボリューム名の変更はコマンド行または GUI のどちらでもできます。

ボリュームの名前を変更する場合は、次のガイドラインを考慮してください。

- 現在使用されているボリュームの名前を変更することはできません。この制限には、マウントされたファイルシステム、`swap`、またはアプリケーションやデータベースのアクティブなストレージとして使用されているボリュームが含まれます。そのため、`metarename` コマンドを使用する前に、名前を変更するボリュームへのすべてのアクセスを停止してください。たとえば、マウントされたファイルシステムをアンマウントします。
- 障害が発生した状態にあるボリュームを交換することはできません。
- ホットスベアの置き換えを使用しているボリュームを交換することはできません。
- 交換は、直接の親子関係を持つボリュームの間でのみ実行できます。
- ログデバイスを交換(または名前変更)することはできません。回避方法として、そのログデバイスを切り離し、目的の名前を持つ別のログデバイスを接続します。
- 交換できるのはボリュームだけです。スライスまたはホットスベアは交換できません。

▼ ボリュームの名前を変更する方法

始める前に [ボリューム名の要件 \(47 ページの「ボリューム名」\)](#)、および [240 ページの「ボリュームの名前の変更の背景情報」](#) を確認してください。

- 1 そのボリュームを使用しているファイルシステムをアンマウントします。

```
# umount /filesystem
```

- 2 ボリユームの名前を変更するには、次のいずれかの方法を使用します。
 - Solaris 管理コンソール内の拡張ストレージツールから、「ボリユーム」を開きます。名前を変更するボリユームを選択します。マウスの右ボタンでアイコンをクリックします。「プロパティ」オプションを選択します。次に、画面の指示に従います。詳細は、オンラインヘルプを参照してください。
 - 次の形式の `metarename` コマンドを使用します。


```
# metarename old-volume-name new-volume-name
```

`old-volume-name` 既存のボリユームの名前を指定します。

`new-volume-name` 既存のボリユームの新しい名前を指定します。

詳細は、[metarename\(1M\)](#) のマニュアルページを参照してください。
- 3 必要に応じて、`/etc/vfstab` ファイルを編集して新しいボリユーム名を参照するようにします。
- 4 ファイルシステムを再マウントします。


```
# mount /filesystem
```

例 20-3 ファイルシステムで使用されているボリユームの名前の変更

次の例では、ボリユーム `d10` が `d100` に変更されます。

```
# umount /home
# metarename d10 d100
d10: has been renamed to d100
  (Edit the /etc/vfstab file so that the file system references the new volume)
# mount /home
```

`d10` にはマウントされたファイルシステムが含まれているため、ボリユームの名前を変更するには、その前にファイルシステムをアンマウントする必要があります。そのボリユームが `/etc/vfstab` ファイル内にエントリを持つファイルシステムで使用されている場合は、新しいボリユーム名を参照するようにそのエントリを変更する必要があります。

たとえば、`/etc/vfstab` ファイルにファイルシステムの次のエントリが含まれている場合は、

```
/dev/md/dsk/d10 /dev/md/rdisk/d10 /docs home 2 yes -
```

そのエントリを次のように変更します。

```
/dev/md/dsk/d100 /dev/md/rdisk/d100 /docs home 2 yes -
```

次に、ファイルシステムを再マウントします。

既存のミラーまたはトランザクションボリュームがある場合は、`metarename -x` コマンドを使用してそのミラーまたはトランザクションボリュームを削除し、基となるボリューム上にデータを保持することができます。トランザクションボリュームの場合は、マスターデバイスがボリューム (RAID-0、RAID-1、または RAID-5 ボリュームのいずれか) であるかぎり、そのボリューム上にデータを保持できます。

構成ファイルの操作

Solaris Volume Manager 構成ファイルには、Solaris Volume Manager の基本的な情報のほか、構成を再構築するために必要なほとんどのデータが含まれています。次の手順は、これらのファイルを操作する方法を示しています。

▼ 構成ファイルを作成する方法

- **Solaris Volume Manager** 環境に該当するすべてのパラメータを定義したら、`metastat -p` コマンドを使用して `/etc/lvm/md.tab` ファイルを作成します。

```
# metastat -p > /etc/lvm/md.tab
```

このファイルには、`metainit` コマンドや `metahs` コマンドで使用されるすべてのパラメータが含まれています。このファイルは、類似したいくつかの環境を設定する必要がある場合や、システム障害が発生したあとに構成を再作成する必要がある場合に使用します。

`md.tab` ファイルの詳細は、326 ページの「`md.tab` ファイルの概要」および `md.tab(4)` のマニュアルページを参照してください。

▼ 構成ファイルから **Solaris Volume Manager** を初期化する方法



注意- この手順は、次の状況で使用します。

- Solaris Volume Manager 構成の完全な消失が発生した場合
- まだ構成が存在せず、保存されている構成ファイルから構成を作成したい場合

場合によっては、状態データベースに保持されていた情報が失われることがあります。たとえば、この消失は、状態データベースの複製がすべて削除されたあとにシステムがリブートされると発生することがあります。状態データベースが失われたあとにボリュームが作成されていないかぎり、`md.cf` または `md.tab` ファイルを使用して Solaris Volume Manager 構成を回復できます。

注-md.cf ファイルには、アクティブなホットスペア上の情報が保持されません。そのため、Solaris Volume Manager 構成が失われたときにホットスペアが使用されていた場合、アクティブなホットスペアを使用していたボリュームは破損する可能性があります。

これらのファイルの詳細は、[md.cf\(4\)](#) および [md.tab\(4\)](#) のマニュアルページを参照してください。

- 1 状態データベースの複製を作成します。
詳細は、[74 ページ](#)の「[状態データベースの複製の作成](#)」を参照してください。
- 2 `/etc/lvm/md.tab` ファイルを作成または更新します。
 - 最後の既知の Solaris Volume Manager 構成を回復しようとしている場合は、`md.cf` ファイルを `/etc/lvm/md.tab` ファイルにコピーします。
 - 保持している `md.tab` ファイルのコピーに基づいて新しい Solaris Volume Manager 構成を作成している場合は、保持されたファイルを `/etc/lvm/md.tab` ファイルにコピーします。
- 3 「新しい」`/etc/lvm/md.tab` ファイルを編集し、次の手順を実行します。
 - 新しい構成を作成するか、またはクラッシュのあとに構成を回復している場合は、ミラーを一方向のミラーとして構成します。例:

```
d80 -m d81 1
d81 1 1 c1t6d0s3
```

ミラーのサブミラーが同じサイズでない場合は、この一方向のミラーには必ず最小のサブミラーを使用してください。そうしないと、データが失われます。
 - 既存の構成を回復しており、Solaris Volume Manager がクリーンな状態で停止された場合は、ミラー構成を多方向のミラーのままにします。例:

```
d70 -m d71 d72 1
d71 1 1 c1t6d0s2
d72 1 1 c1t5d0s0
```
 - デバイスが再初期化されないように、`-k` オプションを使用して RAID-5 ボリュームを指定します。例:

```
d45 -r c1t3d0s5 c1t3d0s3 c1t3d0s4 -k -i 32b
```

詳細については、[metainit\(1m\)](#) のマニュアルページを参照してください。

- 4 次のいずれかの形式の **metainit** コマンドを使用して、変更を確定せずに `/etc/lvm/md.tab` ファイルエントリの構文を確認します。

```
# metainit -n md.tab-entry
```

```
# metainit -n -a
```

metainit コマンドでは、`-n` を使用して実行しているときに作成された可能性のあるデバイスの仮想的な状態が保持されないため、ほかの存在しないボリュームに依存するボリュームを作成すると、そのコマンドが `-n` オプションなしで成功する可能性があったとしても、`-n` でエラーが発生します。

`-n` 実際にデバイスを作成しないように指定します。このオプションは、結果が予期したとおりであることを確認するために使用します。

`md.tab-entry` 初期化するコンポーネントの名前を指定します。

`-a` すべてのコンポーネントを確認するように指定します。

- 5 前の手順で明らかかな問題がなかった場合は、**md.tab** ファイルからボリュームとホットスペアプールを再作成します。

```
# metainit -a
```

`-a` `/etc/lvm/md.tab` ファイル内のエントリをアクティブにするように指定します。

- 6 必要に応じて、**metattach** コマンドを使用して、一方向のミラーを多方向のミラーにします。

```
# mettach mirror submirror
```

- 7 ボリューム上のデータを検証して、構成が正確に再構築されていることを確認します。

```
# metastat
```

Solaris Volume Manager のデフォルト値の変更

Solaris 10 リリースでは、Solaris Volume Manager がボリュームを動的に構成するように拡張されました。`/kernel/drv/md.conf` ファイル内の `nmd` および `md_nsets` パラメータを編集する必要はなくなりました。必要に応じて、新しいボリュームが動的に作成されます。

Solaris Volume Manager 構成の最大値は変更されずに残っています。

- サポートされるボリュームの最大数は 8192 です。
- サポートされるディスクセットの最大数は 32 です。

growfs コマンドを使用したファイルシステムの拡張

UFS ファイルシステムを含むボリュームが拡張された(つまり、容量が追加された)あと、追加された容量を認識するには、そのファイルシステムも拡張する必要があります。growfs コマンドを使用して、ファイルシステムを手動で拡張する必要があります。growfs コマンドは、そのファイルシステムがマウントされている間であっても、ファイルシステムを拡張します。ただし、growfs コマンドの実行中、そのファイルシステムへの書き込みアクセスはできません。

raw デバイスを使用するアプリケーション(データベースなど)は、追加された容量を組み込むための独自の方法を備えています。Solaris Volume Manager には、この機能は用意されていません。

growfs コマンドは、マウントされたファイルシステムを拡張するとき、そのファイルシステムを「書き込みロック」します。ファイルシステムが書き込みロックされる時間の長さは、そのファイルシステムを段階的に拡張することによって短縮できます。たとえば、1G バイトのファイルシステムを 2G バイトに拡張するには、-s オプションを使用して、ファイルシステムを 16M バイトの各段階で拡張できます。このオプションは、各段階での新しいファイルシステムの合計サイズを指定します。

書き込みロック機能のため、拡張中、そのファイルシステムを書き込みアクセスで使用することはできません。書き込みアクセスは透過的に中断され、growfs コマンドがファイルシステムのロックを解除すると再開されます。読み取りアクセスは影響を受けません。ただし、ロックが有効な間、アクセス時間は保持されません。

スライスとボリュームの拡張の背景情報

注 - Solaris Volume Manager ボリュームは拡張できます。ただし、ボリュームのサイズを減らすことはできません。

- ボリュームは、ファイルシステム、アプリケーション、データベースのいずれで使用されていても拡張できます。RAID-0(ストライプおよび連結)ボリューム、RAID-1(ミラー)ボリューム、RAID-5 ボリューム、およびソフトパーティションを拡張できます。
- 既存のファイルシステムを含むボリュームを、そのファイルシステムの使用中に連結できます。ファイルシステムが UFS ファイルシステムであるかぎり、そのファイルシステムを (growfs コマンドを使用して) より大きな容量を満たすように拡張できます。データへの読み取りアクセスを中断することなく、ファイルシステムを拡張できます。
- UFS ファイルシステムの制約のために、ファイルシステムが拡張されたあと、そのサイズを減らすことはできません。

- raw デバイスを使用するアプリケーションやデータベースは、追加された容量を拡張して認識できるようにするための独自の方法を備えている必要があります。Solaris Volume Manager には、この機能は用意されていません。
- RAID-5 ボリュームにコンポーネントが追加された場合、そのコンポーネントはボリュームへの連結になります。新しいコンポーネントにパリティ情報は含まれていません。ただし、新しいコンポーネント上のデータは、そのボリュームに対して実行される全体的なパリティ計算によって保護されます。
- 追加コンポーネントを追加することによってログデバイスを拡張できません。Solaris Volume Manager は、リポート時に追加の容量を自動的に認識するため、growfs コマンドを実行する必要はありません。
- ソフトパーティションは、基となるボリュームまたはスライスから容量を追加することによって拡張できます。その他のすべてのボリュームは、スライスを追加することによって拡張できます。

▼ ファイルシステムを拡張する方法

始める前に 51 ページの「Solaris Volume Manager コンポーネントを作成するための前提条件」を確認してください。

- 1 ファイルシステムに関連付けられたディスク容量を確認します。

```
# df -hk
```

詳細は、df(1M) のマニュアルページを参照してください。

- 2 論理ボリューム上の UFS ファイルシステムを拡張します。

```
# growfs -M /mount-point /dev/md/rdsk/volume-name
```

-M/mount-point 拡張されるファイルシステムのマウントポイントを指定します。

/dev/md/rdsk/volume-name 拡張するボリュームの名前を指定します。

詳細は、次の例および growfs(1M) のマニュアルページを参照してください。

例 20-4 ファイルシステムの拡張

次の例では、マウントされたファイルシステム /home2 を含むボリューム d10 に新しいスライスが追加されます。growfs コマンドは、-M オプションを /home2 としてマウントポイントを指定しており、それが raw ボリューム /dev/md/rdsk/d10 に拡張されます。growfs コマンドが完了すると、ファイルシステムがボリューム全体にまたがります。ファイルシステムを拡張する前とあとに df -hk コマンドを使用して合計ディスク容量を確認できます。

```
# df -hk
Filesystem            kbytes    used  avail capacity  Mounted on
...
/dev/md/dsk/d10       69047    65426      0   100%    /home2
...
# growfs -M /home2 /dev/md/rdisk/d10
/dev/md/rdisk/d10:      295200 sectors in 240 cylinders of 15 tracks, 82 sectors
      144.1MB in 15 cyl groups (16 c/g, 9.61MB/g, 4608 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
   32, 19808, 39584, 59360, 79136, 98912, 118688, 138464, 158240, 178016, 197792,
  217568, 237344, 257120, 276896,
# df -hk
Filesystem            kbytes    used  avail capacity  Mounted on
...
/dev/md/dsk/d10       138703    65426   59407    53%    /home2
...
```

ミラーボリュームの場合は、`growfs` コマンドを常にトップレベルのボリュームで実行します。サブミラーまたはマスターデバイスに容量が追加されたとしても、このコマンドをサブミラーやマスターデバイスで実行しないでください。

RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の概要

Solaris Volume Manager では、RAID-1(ミラー) および RAID-5 ボリューム内のコンポーネントの置き換えや有効化を行うことができます。

Solaris Volume Manager の用語で、コンポーネントの置き換えとは、サブミラーまたは RAID-5 ボリューム内の選択されたコンポーネントをシステム上の使用可能なコンポーネントに置き換える方法を指します。このプロセスは、コンポーネントの物理的な交換とは対照的に、論理的な交換と考えることができます。詳細は、[249 ページ](#)の「[コンポーネントを別の使用可能なコンポーネントに置き換える](#)」を参照してください。

コンポーネントの有効化とは、コンポーネントを「アクティブにする」か、またはそれ自体に置き換える(つまり、コンポーネント名が同じ)ことを指します。詳細は、[249 ページ](#)の「[コンポーネントの有効化](#)」を参照してください。

注- ディスクエラーから回復する場合は、`/var/adm/messages` をスキャンして、どのような種類のエラーが発生したかを確認します。エラーが一時的で、ディスク自体に問題がない場合は、障害が発生したコンポーネントを有効にしてみてください。また、`format` コマンドを使用してディスクをテストすることもできます。

コンポーネントの有効化

次のいずれかの条件が存在する場合は、コンポーネントを有効にすることができます。

- Solaris Volume Manager が物理ドライブにアクセスできない。この問題は、電力損失やドライブケーブルの緩みなどのために発生する可能性があります。この場合は、Solaris Volume Manager がそのコンポーネントを「保守」状態にします。そのドライブに確実にアクセスできるようにしてから (電力の回復やケーブルの再接続など)、ボリューム内のコンポーネントを有効にする必要があります。
- 物理ドライブで、ディスクに関連しない一時的な問題が発生していることが疑われる。「保守」状態にあるコンポーネントを単純に有効にするだけで修復できることがあります。コンポーネントを有効にしても問題が解決されない場合は、次のいずれかを行う必要があります。
 - ディスクドライブを物理的に交換し、そのコンポーネントを有効にします。
 - そのコンポーネントをシステム上の別の使用可能なコンポーネントに置き換えます。

ディスクを物理的に交換した場合は、そのディスクを交換されたディスクと同様にパーティション分割して、使用されている各コンポーネント上に十分な容量を確保してください。

注- 交換されるディスク上にある状態データベースの複製とホットスベアを常に確認してください。エラー状態にある状態データベースの複製はすべて、ディスクを交換する前に削除するようにしてください。次に、コンポーネントを有効にしたあと、同じサイズを使用して状態データベースの複製を再作成します。ホットスベアも同じ方法で処理するようにしてください。

コンポーネントを別の使用可能なコンポーネントに置き換える

既存のコンポーネントを、システムで使用されていない使用可能な別のコンポーネントに置き換えるか、またはスワップする場合は、`metareplace` コマンドを使用します。

このコマンドは、次のいずれかの条件が存在する場合に使用できます。

- ディスクドライブに問題が発生していて、交換用ドライブがない場合。ただし、システム上のほかの場所には使用可能なコンポーネントが存在する。

この方法は、交換が絶対的に必要だが、システムを停止したくない場合にも使用できます。

- 物理ディスクでソフトエラーが表示される場合。
Solaris Volume Manager がミラー/サブミラーまたは RAID-5 ボリュームを「正常」状態で表示している場合でも、物理ディスクでソフトエラーが報告されることがあります。問題のコンポーネントを別の使用可能なコンポーネントに置き換えると、予防保守を実行することができ、ハードエラーの発生が防止される可能性があります。
- パフォーマンスチューニングを行いたい場合。
コンポーネントを評価できる1つの方法として、Solaris 管理コンソール内の拡張ストレージツールから使用可能なパフォーマンスモニタリング機能の使用があります。たとえば、RAID-5 ボリューム内の特定のコンポーネントが、「正常」状態にあるにもかかわらず平均負荷率が高くなっていることを確認できる場合があります。ボリューム上の負荷を分散させるために、そのコンポーネントを使用頻度の低いディスクのコンポーネントに置き換えることができます。このタイプの置き換えは、そのボリュームへのサービスを中断することなくオンラインで実行できます。

「保守」状態と「最後にエラー」状態

RAID-1 または RAID-5 ボリューム内のコンポーネントでエラーが発生すると、Solaris Volume Manager はそのコンポーネントを「保守」状態にします。「保守」状態にあるコンポーネントへのそれ以上の読み取りまたは書き込みは実行されません。

場合によっては、コンポーネントが「最後にエラー」状態になることがあります。RAID-1 ボリュームの場合、これは通常1面ミラーで発生します。ボリュームでエラーが発生します。しかし、読み取り元となる冗長なコンポーネントが存在しません。RAID-5 ボリュームの場合、これは1つのコンポーネントが「保守」状態になり、別のコンポーネントで障害が発生したあとに起こります。障害が発生した2つ目のコンポーネントが「最後にエラー」状態になります。

RAID-1 ボリュームまたは RAID-5 ボリュームのどちらかに「最後にエラー」状態のコンポーネントがある場合は、その「最後にエラー」のマークが付いたコンポーネントへの入出力が引き続き試みられます。この入出力の試みが起こるのは、Solaris Volume Manager の側から見た場合に、「最後にエラー」状態のコンポーネントにデータの最後の適切なコピーが含まれているからです。「最後にエラー」状態にあるコンポーネントを含むボリュームは正常なデバイス(ディスク)のように動作し、アプリケーションに入出力エラーを返します。通常は、この時点で、一部のデータが失われています。

同じボリューム内のほかのコンポーネントに関する以降のエラーは、ボリュームのタイプに応じて異なった方法で処理されます。

RAID-1 ボリューム RAID-1 ボリュームは、「保守」状態にある多数のコンポーネントを許容し、引き続き読み取ったり書き込んだりできる可

能性があります。コンポーネントが「保守」状態にある場合、データは失われていません。コンポーネントを任意の順序で安全に置き換えるか、または有効にすることができません。コンポーネントが「最後にエラー」状態にある場合は、まず「保守」状態にあるコンポーネントを置き換えるまで、そのコンポーネントを置き換えることができません。「最後にエラー」状態にあるコンポーネントの置き換えまたは有効化は通常、一部のデータが失われていることを示します。修復したあと、必ずそのミラー上のデータを検証してください。

RAID-5 ボリューム RAID-5 ボリュームは、「保守」状態にある1つのコンポーネントを許容できます。データを失うことなく、「保守」状態にある1つのコンポーネントを安全に置き換えることができます。別のコンポーネントでエラーが発生した場合は、そのコンポーネントが「最後にエラー」状態になります。この時点で、RAID-5 ボリュームは読み取り専用デバイスになります。RAID-5 ボリュームの状態が安定し、データ損失の可能性が低くなるように、何らかの種類のエラー回復を実行する必要があります。RAID-5 ボリュームが「最後にエラー」状態に達した場合は、データが失われている可能性があります。修復したあと、必ずその RAID-5 ボリューム上のデータを検証してください。

常に、まず「保守」状態にあるコンポーネントを置き換え、そのあとで「最後にエラー」状態にあるコンポーネントを置き換えてください。コンポーネントが置き換えられ、再同期されたら、`metastat` コマンドを使用してその状態を確認します。次に、データを検証します。

RAID-1 および RAID-5 ボリューム内のコンポーネントの置き換えと有効化の背景情報

RAID-1 ボリュームまたは RAID-5 ボリューム内のコンポーネントを置き換える場合は、次のガイドラインに従ってください。

- 常に、まず「保守」状態にあるコンポーネントを置き換え、そのあとで「最後にエラー」状態にあるコンポーネントを置き換えてください。
- コンポーネントが置き換えられ、再同期されたら、`metastat` コマンドを使用してボリュームの状態を確認します。次に、データを検証します。「最後にエラー」状態にあるコンポーネントの置き換えまたは有効化は通常、一部のデータが失われていることを示します。修復したあと、必ずそのボリューム上のデータを検証してください。UFS の場合は、`fsck` コマンドを実行して「メタデータ」(ファイルシステムの構造)を検証します。次に、実際のユーザーデータ

を確認します。(実際には、各ユーザーが自分のファイルを検査する必要があります。)データベースやその他のアプリケーションは、自身の内部データ構造を検証するための独自の方法を備えている必要があります。

- コンポーネントを置き換える場合は、状態データベースの複製とホットスペアを常に確認してください。エラー状態にある状態データベースの複製はすべて、物理ディスクを交換する前に削除するようにしてください。状態データベースの複製は、そのコンポーネントを有効にする前に戻すようにしてください。ホットスペアにも同じ手順が適用されます。
- RAID-5 ボリュームのコンポーネントの置き換え中に、データは2つの方法のどちらかで回復されます。データは、現在使用されているホットスペアから回復されるか、またはホットスペアが使用されていない場合は RAID-5 パリティを使用して回復されます。
- RAID-1 ボリュームのコンポーネントを置き換えると、Solaris Volume Manager は、ボリュームの残りの部分を使用して新しいコンポーネントの再同期を自動的に開始します。再同期が完了すると、置き換えられたコンポーネントは読み取り/書き込み可能になります。障害が発生したコンポーネントがホットスペアのデータで置き換えられると、そのホットスペアが「使用可能」状態になり、ほかのホットスペアの置き換えに使用できるようになります。
- 新しいコンポーネントには、古いコンポーネントを置き換えるだけの十分な大きさが必要です。
- 予防策として、「最後にエラー」のデバイスを置き換える前に、すべてのデータをバックアップしてください。

Solaris Volume Manager のベストプラクティス

この章では、Solaris Volume Manager を使用した実際のストレージのシナリオからの一般的なベストプラクティス情報を提供します。この章では、標準的な構成について説明したあと、その分析を行い、同じニーズを満たすための推奨される(「ベストプラクティス」)の構成を示します。

この章の内容は次のとおりです。

- 253 ページの「小規模なサーバーの配備」
- 255 ページの「ネットワーク接続されたストレージデバイスによる Solaris Volume Manager の使用」

小規模なサーバーの配備

分散コンピューティング環境では多くの場合、複数の場所に類似または同一のサーバーを配備する必要があります。これらの環境には、ISP、地理的に分散した営業所、電気通信サービスプロバイダなどがあります。分散コンピューティング環境内のサーバーは、次のサービスのうちのいくつかを提供する可能性があります。

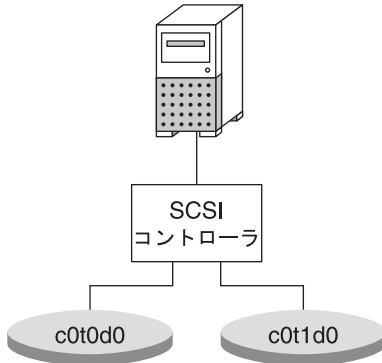
- ルーターまたはファイアウォールサービス
- 電子メールサービス
- DNS キャッシュ
- Usenet (Network News) サーバー
- DHCP サービス
- さまざまな場所で最適に提供されるその他のサービス

これらの小規模なサーバーには、いくつかの共通した特性があります。

- 高信頼性の要件
- 高可用性の要件
- 日常的なハードウェアとパフォーマンスの要件

開始点として、1つのSCSIバスと2つの内蔵ディスクを備えたNetraサーバーを考えてみます。この汎用的な構成は、分散したサーバーのための良い開始点になります。Solaris Volume Managerを使用すると、スライスの一部またはすべてを容易にミラー化できるため、ディスク障害からの保護に役立つ冗長ストレージを実現できます。この小規模なシステム構成の例については、次の図を参照してください。

図21-1 小規模なシステム構成



この構成には、ルート (/)、/usr、swap、/var、および/export ファイルシステムに加え、状態データベースの複製のミラーが含まれている可能性があります(ディスクごとに1つ)。そのため、ミラーのどちらの側に障害が発生しても、必ずしもシステム障害を引き起こすわけではありません。また、最大5つの個別の障害も許容できる可能性があります。ただし、このシステムが、ディスクやスライスの障害から十分に保護されているわけではありません。さまざまな潜在的な障害によって完全なシステム障害が発生し、オペレータの介入が必要になる可能性があります。

この構成は、致命的なディスク障害からのある程度の保護を提供するのに役立ちますが、次の重大なシングルポイント障害が発生することがあります。

- 1つのSCSIコントローラが、潜在的な障害ポイントになります。このコントローラに障害が発生した場合は、システムが停止し、部品交換待ちの状態になります。
- 2つのディスクでは、状態データベースの複製の十分な分散が実現されません。多数決コンセンサスアルゴリズムでは、システムが引き続き動作するには、状態データベースの複製の半分が使用できる必要があります。このアルゴリズムではまた、リブートには、半分に加えて1つの複製も必要になります。そのため、状態データベースの複製が各ディスク上に1つ存在し、その複製を含む1つのディスクまたはスライスに障害が発生した場合、システムはリブートできません。その結果、ミラー化されたルート (/) ファイルシステムは効果的でなくなります。状態データベースの複製が各ディスク上に2つ以上存在する場合、1つのスライス障害は問題でなくなる可能性があります。ただし、ディスク障害によって、リブートは引き続き阻止されます。各ディスク上に異なる数の複製が存

在する場合、1つのディスクには半分を超える複製が存在し、1つのディスクには半分より少ない複製が存在します。少ない複製を含むディスクに障害が発生した場合、システムはリブートして続行することができます。ただし、多くの複製を含むディスクに障害が発生した場合、システムではただちにパニックが発生しません。

「ベストプラクティス」のアプローチとして、コントローラを1つおよびハードドライブを1つ追加することによって、構成を変更します。結果として得られる構成は復元性ははるかに向上します。

ネットワーク接続されたストレージデバイスによる Solaris Volume Manager の使用

Solaris Volume Manager は、ネットワーク接続されたストレージデバイス、特に構成可能な RAID レベルと柔軟なオプションを提供するデバイスとともに適切に動作します。通常は、Solaris Volume Manager とこのようなデバイスを組み合わせることにより、どちらかの製品だけより優れたパフォーマンスと柔軟性が得られます。

一般に、Solaris Volume Manager の RAID-5 ボリュームは、冗長性を提供するハードウェアストレージデバイス (RAID-1 ボリュームや RAID-5 ボリュームなど) 上には確立しないでください。非常にまれな状況でないかぎり、パフォーマンスが低下します。また、冗長性または高可用性の点から見た場合の利点も、非常に少なくなります。

これに対して、基となるハードウェアストレージデバイスを RAID-5 ボリュームで構成することは非常に効果的です。これにより、Solaris Volume Manager ボリュームのための適切な基礎が提供されます。ハードウェア RAID-5 は、Solaris Volume Manager の RAID-1 ボリューム、ソフトパーティション、またはその他のボリュームに追加の冗長性を提供します。

注- 類似したソフトウェアとハードウェアのデバイスを構成しないでください。たとえば、ハードウェア RAID-1 デバイスの上にソフトウェア RAID-1 ボリュームを構築しないでください。ハードウェアとソフトウェアで類似したデバイスを構成すると、信頼性に関しては何も得られることなく、パフォーマンスが低下します。

基となるハードウェアストレージデバイス上に構築された Solaris Volume Manager の RAID-1 ボリュームは、RAID-1+0 ではありません。Solaris Volume Manager は、RAID-1+0 機能を提供できるほど十分に基となるストレージを認識できません。

ハードウェア RAID-5 デバイス上に順番に構築された Solaris Volume Manager の RAID-1 ボリュームの上にソフトパーティションを構成することは、非常に柔軟で、復元性の高い構成です。

トップダウンボリューム作成 (概要)

この章では、Solaris Volume Manager の「トップダウン」ボリューム作成の概念について説明します。

この章では、次の内容について説明します。

- 257 ページの「トップダウンボリューム作成の概要」
- 258 ページの「ディスクセットによるトップダウンボリューム作成の実装」
- 259 ページの「トップダウンボリューム作成のプロセス」
- 260 ページの「トップダウンボリューム作成に使用できるディスクの判別」

関連タスクの実行の詳細については、第 23 章「ボリュームのトップダウン作成(タスク)」を参照してください。

トップダウンボリューム作成の概要

トップダウンボリューム作成により、`metassist` コマンドを使用して、Solaris Volume Manager ボリューム構成を自動的に作成することができます。ディスクのパーティション分割、RAID-0 ボリュームの作成(サブミラーとして)、ホットスペアプールおよびホットスペアの作成、および最終的なミラーの作成のプロセスを手動で実行する必要がなくなります。代わりに、`metassist` コマンドを発行して、ボリュームを作成できます。Solaris Volume Manager によって残りの手順が自動的に実行されます。

`metassist` コマンドを使用すると、1つのコマンドで Solaris Volume Manager のボリューム構成を作成できます。「サービス品質」の点から、ボリュームの特性を指定できます。サービス品質の特性とは、ボリュームで使用するハードウェアコンポーネントを指定することなく、`metassist` コマンドへの入力を使用して、以下を指定できることを意味します。

- ボリュームサイズ
- 冗長性のレベル(データのコピー数)

- ボリュームへのデータパス数
- 障害回復(ボリュームがホットスペアプールに関連付けられているかどうか)

ボリュームは、コマンド行オプションや、コマンド行に指定する入力ファイルに、サービス品質によって指定することができます。

場合によっては、ボリューム特性やボリューム作成の制約を具体的に定義することが重要になります。そのような場合は、次の特性も指定できます。

- ボリュームタイプ (RAID-0 (連結) または RAID-0 (ストライプ) ボリュームなど)。
- 特定のボリュームで使用するコンポーネント。
- 使用できるコンポーネントまたは使用できないコンポーネント。
- 使用するコンポーネントの数。
- 作成中のボリュームのタイプに固有の詳細。詳細には、ストライプ、ミラーの読み取りポリシー、および類似の特性が含まれます。

ボリュームの名前、サイズ、コンポーネントを詳細に指定したい場合は、入力ファイルを使います。入力ファイルには、ボリュームリクエストファイルとボリューム仕様ファイルが含まれます。入力ファイルの使用の詳細については、[259 ページの「トップダウンボリューム作成のプロセス」](#)を参照してください。

最後に、`metassist` コマンドを特定のディスクまたはパスに使用する (または使用しない) ように制限することができます。

ディスクセットによるトップダウンボリューム作成の実装

`metassist` コマンドは Solaris Volume Manager ディスクセットを使用して、トップダウンボリューム作成のボリュームと使用可能なディスクを管理します。トップダウンボリューム作成プロセスでは、構成要素として使用されるすべてのディスクが、ディスクセットに含まれているか、ディスクセットに追加可能である必要があります。トップダウン作成プロセスを使用すると、さまざまなディスクセットにボリュームを作成できます。ただし、使用できるディスクやコンポーネントは、ディスクセットの機能によって制限されます。

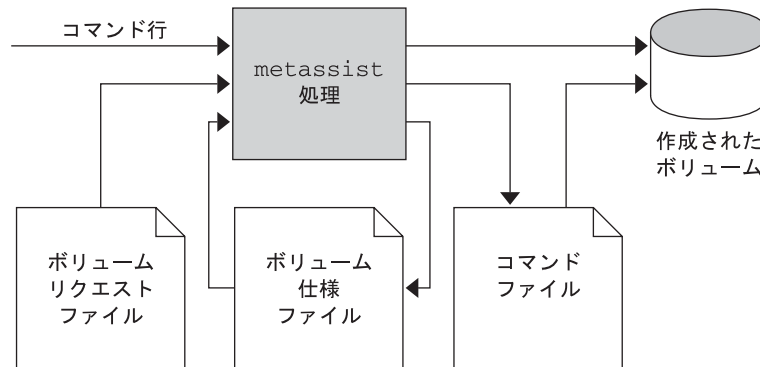
トップダウンボリューム作成のプロセス

トップダウンボリューム作成プロセスでは、次のプロセスを提供して、柔軟性を実現しています。

- 必要な制約を指定でき、コマンドが完了した時に必要なボリュームが作成される、完全に自動化されたエンドツーエンドのプロセス
- ブレークポイントでXML ベースのファイルに書き込むことができるより詳細なプロセス

次の図に、コマンド行入力と入力ファイルに基づいて、`metassist` コマンドがどのようにエンドツーエンドプロセスをサポートするかを示します。さらに、図には、`metassist` コマンドで部分処理サポートする方法も示します。これにより、ファイルベースのデータが得られ、ボリュームの特性をチェックできます。

図 22-1 トップダウンボリューム作成の処理のオプション



自動のボリューム作成アプローチでは、コマンド行を使用して、必要なサービス品質特性を指定します。`metassist` コマンドは、リクエストされたボリュームを自動的に作成します。例:

```
# metassist create -s storagepool -S 10Gb
```

このコマンドは `storagepool` ディスクセットに 10G バイトのサイズのストライプボリュームを作成します。コマンドは、`storagepool` ディスクセットに存在する使用可能なストレージを使用します。

または、ボリュームリクエストファイルを使用して、ボリュームの特性を定義することができます。次に、`metassist -F request-file` コマンドを使用して、それらの特性を持つボリュームを作成できます。

`metassist -d` コマンドを使用すると、ボリューム仕様ファイルを作成できます。このファイルを使用して、目的の実装を評価し、必要に応じてファイルを編集できます。その後、このボリューム仕様ファイルは、`metassist` コマンドの入力として使用してボリュームを作成できます。

最後に、`metassist -c` コマンドを使用して、コマンドファイルを作成できます。コマンドファイルは `metassist` コマンドで指定された Solaris Volume Manager のデバイス構成を実装するシェルスクリプトです。繰り返しボリュームの作成にこのファイルを使用し、適宜、ファイルを編集できます。

`metassist` コマンドを使用してこれらのファイルを作成すると、`metassist` コマンドが何を実行するか、どのように決定するかを学ぶことができます。この情報は、以下のようなトラブルシューティングに役立ちます。

- ボリュームがなぜ特定の方法で作成されたのか。
- ボリュームがなぜ作成されなかったのか。
- `metassist` コマンドはどのようなボリュームを作成するのか (ボリュームを実際に作成せずに)。

トップダウンボリューム作成に使用できるディスクの判別

`metassist` コマンドはディスクをチェックし、使用されていないように見えるディスクを判別します。このコマンドは、使用できるディスクを慎重に判別しようとしません。使用中のディスクやスライスは、`metassist` コマンドで使用できません。`metassist` コマンドは次をチェックします。

- 他のディスクセットで使用されているディスク
- マウントされているスライス
- ファイルシステムスーパーブロックをもつスライス (マウント可能なファイルシステム)
- ほかの Solaris Volume Manager のボリュームで使用されているスライス

これらのいずれかの条件を満たしているスライスは、トップダウンボリューム作成に使用できません。

ボリュームのトップダウン作成(タスク)

この章では、`metassist` コマンドを使用して行う、Solaris Volume Manager のボリュームトップダウン作成に関連するタスクについて説明します。

この章の内容は次のとおりです。

- 261 ページの「ボリュームのトップダウン作成(タスクマップ)」
- 262 ページの「ボリュームをトップダウン作成するための前提条件」
- 263 ページの「ボリュームの自動作成」
- 267 ページの「`metassist` コマンドによるファイルベースのデータ処理」
- 274 ページの「`metassist` コマンドのデフォルト動作の変更」

ボリュームのトップダウン作成に関連する概念については、第 22 章「トップダウンボリューム作成(概要)」を参照してください。

ボリュームのトップダウン作成(タスクマップ)

次のタスクマップは、`metassist` コマンドを使用して Solaris Volume Manager のボリュームをトップダウンで作成するために必要な手順を示します。このコマンドでは、QoS(サービス品質)特性に基づいてボリュームを指定でき、また、1つのコマンドで、レイヤー型のボリュームセットを作成できます。

タスク	説明	参照先
ボリュームを自動的に作成する	metassist コマンドを使って、1 つまたは複数の Solaris Volume Manager のボリュームを作成できます。 また、ボリュームの作成過程で metassist コマンドから出力されるトラブルシューティングや診断のための情報量を制御します。	263 ページの「ボリュームの自動作成」 263 ページの「出力の詳細度指定によるボリューム作成分析」
コマンドファイルを作成する	metassist コマンドを使って、指定したボリュームを生成するためのシェルスクリプトを作成します。	272 ページの「metassist コマンドによるボリューム構成ファイルの作成」
シェルスクリプトを使用してボリュームを作成する	前の手順において metassist コマンドで生成したシェルスクリプトを使用して、Solaris Volume Manager のボリュームを作成します。	271 ページの「metassist コマンドで作成されたシェルスクリプトによるボリュームの作成」
ボリューム構成ファイルを作成する	作成するボリュームの特性を定義したボリューム構成ファイルを作成します。	272 ページの「metassist コマンドによるボリューム構成ファイルの作成」
ボリュームデフォルトファイルを変更する	デフォルトのボリューム特性を設定して、metassist コマンドの動作をカスタマイズします。	275 ページの「ボリュームデフォルトファイルの変更」

ボリュームをトップダウン作成するための前提条件

metassist コマンドを使ってボリュームやボリューム構成を自動的に作成するためには、Solaris Volume Manager 構成が正常に動作していなければなりません。始める前に、次のものがが必要です。

- スーパーユーザーのアクセス権または役割に基づくアクセス制御 (RBAC) の同等の役割。詳細は、『Oracle Solaris の管理: 基本管理』の「スーパーユーザー (root) になるか役割を引き受ける」を参照してください。
- 使用するシステム上に適切に分散された状態データベースの複製。状態データベースの複製の詳細は、67 ページの「Solaris Volume Manager 状態データベースと複製について」を参照してください。
- ボリュームの作成に使用できるディスク。metassist コマンドではディスクセットを使用してストレージを管理します。metassist コマンドで新しいボリュームを作成するには、まったく未使用のディスク (または既存のディスク

セット)を使用する必要があります。ディスクの可用性については、[260 ページ](#)の「[トップダウンボリューム作成に使用できるディスクの判別](#)」を参照してください。

これらの最小要件に加えて、`/etc/inetd.conf` ファイル内で Solaris Volume Manager の RPC(`rpc.metad`、`rpc.metamhd`、および `rpc.metamedd`)を無効にはなりません。デフォルトでは、これらのデーモンは起動するように構成されています。Solaris Volume Manager が共有ディスクセットを使用できるように、これらのデーモンを引き続き有効にしておく必要があります。

ボリュームの自動作成

`metassist` コマンドを使用すると、QoS(サービス品質)の条件に基づいて、Solaris Volume Manager のボリュームやボリュームセットを作成できます。Solaris Volume Manager ではこれまで、ボリュームを作成するために一連のコマンドが必要でしたが、`metassist` コマンドによって、1つのコマンドでボリュームを作成できるようになりました。

`metassist` コマンドを使用すると、RAID-1(ミラー)ボリュームを直接作成できます。したがって、RAID-1(ミラー)ボリュームのコンポーネントとして使用するサブミラー(連結方式またはストライプ方式)を先に作成する必要はありません。

出力の詳細度指定によるボリューム作成分析

`metassist` コマンドを実行する際には、出力の詳細度を指定できます。出力が詳細になれば、それだけ問題の診断に役立ちます。たとえば、あるディスクがボリュームの作成になぜ選択されたのか、あるいは選択されなかったのかを判別したり、特定のコマンドがなぜ失敗したのかを判別したりすることが容易になります。出力の詳細度を下げれば、ユーザーに不必要な情報の出力を減らすことができます。

出力の詳細度を指定すると、`metassist` コマンドが何を行い、どのように決定を下すかを理解できます。この情報は、次のようなトラブルシューティングに役立ちます。

- ボリュームがなぜ特定の 방법으로作成されたのか。
- ボリュームがなぜ作成されなかったのか。
- `metassist` コマンドはどのようなボリュームを作成するのか(ボリュームを実際に作成せずに)。

▼ metassist コマンドを使用して RAID-1 (ミラー) ボリュームを作成する方法

始める前に [262 ページの「ボリュームをトップダウン作成するための前提条件」](#)を確認してください。

- 1 ボリュームの作成に使用するストレージを特定します。
 ストレージを明示的に指定しなかった場合、システム上の未使用のストレージを Solaris Volume Manager が特定し、必要に応じて使用します。ストレージを指定する場合は、広義(たとえば、コントローラ 1 のすべてのストレージ)または狭義(たとえば、c1t4d2 は使用し、c1t4d1 は使用しない)に指定でき、そのストレージが Solaris Volume Manager で使用されます。
- 2 タスクに応じて、**metassist** コマンドと適切なオプションを使用します。
 - コマンド行からボリュームを作成するには、次の形式の **metassist** コマンドを使用します。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... -S size -v verbosity
```

<code>create</code>	ボリュームを作成するために使用するサブコマンドです。
<code>-s diskset-name</code>	ボリュームに使用するディスクセットの名前を指定します。
<code>-f</code>	ボリュームとホットスペアを対応付けることを指定します。
<code>-r redundancy</code>	作成する冗長レベル(データコピー数)を指定します。
<code>-a device1, device2...</code>	ボリュームの作成に使用できるデバイスを指定します。
<code>-S size</code>	作成するボリュームのサイズを KB(キロバイト)、MB(メガバイト)、GB(ギガバイト)、または TB(テラバイト)単位で指定します。
<code>-v verbosity</code>	出力の詳細度を指定します。指定できる値の範囲は 0(出力がほとんどない)から 2(出力が多い)です。デフォルトレベルは 1(中程度の出力)です。

- ボリュームの特性を指定する入力ファイルを使ってボリュームを作成するには、次のいずれかの形式の **metassist** コマンドを使用します。

```
# metassist create [-v n] [-c] -F config_file
# metassist create [-v n] [-c | -d] -F request_file
```

<code>-c</code>	指定のボリューム構成または生成されたボリューム構成を実装するコマンドスクリプトを出力するように指定します。コマンドスクリプトは実行されず、処理はこの段階で終了します。
-----------------	---

- d** 指定のボリュームリクエストまたは生成されたボリュームリクエストを満たすボリューム構成を出力するように指定します。コマンドスクリプトは生成も実行もされず、処理はこの段階で終了します。
- F *config_file* | *request_file*** 処理対象のボリュームリクエストファイルまたはボリューム構成ファイルを指定します。*config_file* または *request_file* の位置にダッシュ (-) を指定した場合、ファイルは標準入力から読み込まれます。入力ファイルがボリューム構成ファイルである場合、**-d** オプションは指定できません。
- ボリューム構成ファイルには、作成するボリュームの詳細な構成情報が記述されています。一方、ボリュームリクエストファイルには、作成するボリュームの特性が記載されています。詳細は、[volume-config\(4\)](#) [volume-config\(4\)](#) および [volume-request\(4\)](#) のマニュアルページを参照してください。
- v *verbosity*** 出力の詳細度を指定します。指定できる値の範囲は 0 (出力がほとんどない) から 2 (出力が多い) です。デフォルトレベルは 1 (中程度の出力) です。

詳細は、次の例と [metassist\(1M\)](#) のマニュアルページを参照してください。

3 ボリュームの作成後、新しいボリュームを表示します。

```
# metastat -s diskset-name
```

例 23-1 metassist コマンドを使用して 2 面ミラーを作成する

次の例では、容量が 10M バイトの 2 面ミラーを作成する方法を示します。metassist コマンドは、未使用のディスクを特定し、これらのディスクを使ってできるだけ条件の良いミラーを作成します。**-s myset** 引数で、myset ディスクセットにボリュームを作成することを指定します。ディスクセットは必要に応じて作成されます。

```
# metassist create -s myset -r 2 -S 10mb
```

例 23-2 metassist コマンドを使用して 2 面ミラーとホットスペアを作成する

次の例では、metassist コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。**-f** オプションで障害耐性を指定します。

```
# metassist create -s myset -f -r 2 -S 10mb
```

例 23-3 metassist コマンドを使用して特定のコントローラにストライプを作成する

次の例では、metassist コマンドを使用して、コントローラ 1 上の使用可能なディスクでストライプを作成する方法を示します。-a オプションで、使用可能なコントローラを指定します。

```
# metassist create -s myset -a c1 -S 10mb
```

例 23-4 metassist コマンドの出力の詳細度を指定する

次の例では、metassist コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。-f オプションで障害耐性を指定します。最後の引数 (-v 2) では、最大の詳細度である 2 を指定しているため、metassist コマンドの実行結果がもっとも詳細に出力されます。

```
# metassist create -s myset -f -r 2 -S 10mb -v 2
Scanning system physical device configuration...
```

```
These HBA/Controllers are known:.
```

```
  c0                               /pci@1f,0/pci@1,1/ide@3
  c1                               /pci@1f,0/pci@1/pci@2/SUNW,isp2@4
```

```
These disks are known:
```

```
  c0t0d0          id1,dad@AST34342A=_____VG97101
  c1t1d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L88P000021097XNL
  c1t2d0          id1,sd@SSEAGATE_ST39102LCSUN9.0GLJW22867000019171JDF
  c1t3d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L7RV00007108TG0H
  c1t4d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0LDFR000021087R1T
  c1t5d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L0M200002109812L
  c1t6d0          id1,sd@SSEAGATE_ST39204LCSUN9.0G3BV0L8K8000021087R0Z
```

```
.
.
.
```

```
(output truncated)
```

次の例では、metassist コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。-f オプションで障害耐性を指定します。最後の引数 (-v 0) では、最小の詳細度である 0 を指定しているため、コマンドの実行時に出力はほとんどありません。

```
# metassist create -s myset -f -r 2 -S 10mb -v 0
myset/hsp000: Hotspare pool is setup
myset/hsp000: Hotspare is added
myset/d2: Concat/Stripe is setup
myset/d1: Concat/Stripe is setup
```

```
myset/d0: Mirror is setup  
myset/d0: submirror myset/d1 is attached
```

例 23-5 入力ファイルを使ってボリュームを作成する

次の例では、metassist コマンドで入力ファイルを使用してボリュームを作成する方法を示します。

```
# metassist create -F request.xml
```

metassist コマンドで入力ファイルを使用する方法については、[267 ページ](#)の「[metassist コマンドによるファイルベースのデータ処理](#)」を参照してください。

metassist コマンドによるファイルベースのデータ処理

metassist コマンドを使用すると、ボリューム特性の評価に使用するファイル、または実際にボリュームを作成するために使用するファイルを作成できます。

metassist コマンドによるコマンドファイル (シェルスクリプト) の作成

metassist コマンドに `-c` 引数を指定すると、ボリューム構成の作成に使用できるコマンドを含んだ Bourne シェルスクリプトが生成されます。この方法を使用すれば、ボリュームを実際に作成する前にコマンドを確認でき、必要に応じてスクリプトを微調整することもできます。

▼ metassist コマンドを使用してコマンドファイル (シェルスクリプト) を作成する方法

始める前に [262 ページ](#)の「[ボリュームをトップダウン作成するための前提条件](#)」を確認してください。

- 1 ボリュームの作成に使用するストレージを特定します。
ストレージを明示的に指定しなかった場合、システム上の未使用のストレージを Solaris Volume Manager が特定し、必要に応じて使用します。ストレージを指定する場合は、広義 (たとえば、コントローラ 1 のすべてのストレージ) または狭義 (たとえば、c1t4d2 は使用し、c1t4d1 は使用しない) に指定でき、そのストレージが Solaris Volume Manager で使用されません。

- 2 タスクに応じて、**metassist** コマンドと適切なオプションを使用します。
 -c オプションを使用して、ボリュームを実際には作成しないことを指定します。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... \  
-S size -v verbosity [-c]
```

create	ボリュームを作成するために使用するサブコマンドです。
-s <i>diskset-name</i>	ボリュームに使用するディスクセットの名前を指定します。
-f	ボリュームとホットスペアを対応付けることを指定します。
-r <i>redundancy</i>	作成する冗長レベル (データコピー数) を指定します。
-a <i>device1, device2...</i>	ボリュームの作成に使用できるデバイスを指定します。
-S <i>size</i>	作成するボリュームのサイズを KB (キロバイト)、MB (メガバイト)、GB (ギガバイト)、または TB (テラバイト) 単位で指定します。
-v <i>verbosity</i>	出力の詳細度を指定します。指定できる値の範囲は 0 (出力がほとんどない) から 2 (出力が多い) です。デフォルトレベルは 1 (中程度の出力) です。
-c	ボリュームを実際には作成しないことを指定します。代わりに、指定した構成を作成するためのシェルスクリプトが、標準出力に送信されます。

注 --c 引数によって要求されたシェルスクリプトは標準出力に送信され、**metassist** コマンドの他の出力は標準エラーに送信されます。出力ストリームは自由にリダイレクトできます。

詳細は、次の例と **metassist(1M)** のマニュアルページを参照してください。

例 23-6 metassist コマンドによるコマンドファイル(シェルスクリプト)の作成

次の例では、**metassist** コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。-f オプションで障害耐性を指定します。最後の引数 (-c) で、ボリュームを実際には作成しないことを指定します。代わりに、指定した構成を作成するためのシェルスクリプトが、標準出力に送信されます。

```
# metassist create -s myset -f -r 2 -S 10mb -c  
(output truncated)  
.  
.  
.
```

```
Volume request completed successfully.
#!/bin/sh

#
# Environment
#

# Amend PATH
PATH="/usr/sbin:/usr/bin:$PATH"
export PATH

# Disk set name
diskset='myset'

#
# Functions
#

# Echo (verbose) and exec given command, exit on error
execho () {
    test -n "$verbose" && echo "$@"
    "$@" || exit
}

# Get full /dev/rdisk path of given slice
fullpath () {
    case "$1" in
        /dev/dsk/*|/dev/did/dsk/*) echo "$1" | sed 's/dsk/rdisk/' ;;
        /*) echo "$1" ;;
        *) echo /dev/rdisk/"$1" ;;
    esac
}

# Run fmthard, ignore partboot error, error if output
fmthard_special () {
    ignore='Error writing partboot'
    out='fmthard "$@" 2>&1'
    result=$?
    echo "$out" |
    case "$out" in
        *"$ignore"*) grep -v "$ignore"; return 0 ;;
        '') return "$result" ;;
        *) cat; return 1 ;;
    esac >&2
}

#
# Main
#

# Verify root
if [ "$(id | sed 's/^[^()*(\[[^)]*\].*/\1/'" != root ]
then
    echo "This script must be run as root." >&2
    exit 1;
fi

# Check for verbose option
```

```
case "$1" in
    -v) verbose=1 ;;
    *) verbose= ;;
esac

# Does the disk set exist?
if metaset -s "$diskset" >/dev/null 2>&1
then
    # Take control of disk set
    execho metaset -s "$diskset" -t
else
    # Create the disk set
    autotakeargs=
    /usr/sbin/clinfo || autotakeargs='-A enable'
    execho metaset -s "$diskset" $autotakeargs -a -h 'uname -n | cut -f1 -d.'
fi

# Format slices
execho fmthard_special -d 7:0:0:0:0 'fullpath c1t3d0s7'
execho fmthard_special -d 7:0:0:0:0 'fullpath c1t6d0s7'
execho fmthard_special -d 7:0:0:0:0 'fullpath c1t4d0s7'

# Add disks to set
execho metaset -s "$diskset" -a c1t3d0
execho metaset -s "$diskset" -a c1t6d0
execho metaset -s "$diskset" -a c1t4d0

# Format slices
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath c1t3d0s0'
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath c1t6d0s0'
execho fmthard_special -d 0:4:0:10773:17649765 'fullpath c1t4d0s0'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath c1t3d0s1'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath c1t4d0s1'
execho fmthard_special -d 1:4:0:17660538:21546 'fullpath c1t6d0s1'

# Does hsp000 exist?
metahs -s "$diskset" -i hsp000 >/dev/null 2>&1 || {
    # Create hsp hsp000
    execho metainit -s "$diskset" hsp000
}

# Add slices to hsp000
execho metahs -s "$diskset" -a hsp000 c1t3d0s1

# Create concat d2
execho metainit -s "$diskset" d2 1 1 c1t4d0s1

# Associate concat d2 with hot spare pool hsp000
execho metaparam -s "$diskset" -h hsp000 d2

# Create concat d1
execho metainit -s "$diskset" d1 1 1 c1t6d0s1

# Associate concat d1 with hot spare pool hsp000
execho metaparam -s "$diskset" -h hsp000 d1

# Create mirror d0
execho metainit -s "$diskset" d0 -m d2 1
```

```
execho metattach -s "$diskset" d0 d1
#
```

例 23-7 metassist コマンドを使用してコマンドファイル(シェルスクリプト)を保存する

次の例では、metassist コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。-f オプションで障害耐性を指定します。最後の引数(-c)で、ボリュームを実際には作成しないことを指定します。代わりに、指定した構成を作成するためのシェルスクリプトが、標準出力に送信されます。コマンドの最後で、標準出力をリダイレクトして /tmp/metassist-shell-script.sh シェルスクリプトを作成します。これはあとで、指定したボリュームを作成するために使用できます。

```
# metassist create -s myset -f -r 2 -S 10mb -c > \
/tmp/metassist-shell-script.sh
```

metassist コマンドで作成されたシェルスクリプトによるボリュームの作成

metassist コマンドでシェルスクリプトを作成したら、このスクリプトを使ってボリュームを作成できます。ボリュームは、シェルスクリプトの作成時に指定したとおり作成されます。



注意-metassist コマンドで作成したコマンドスクリプトは、スクリプトを作成したシステムのその時点の構成に大きく依存しています。このスクリプトを別のシステムで使用したり、このスクリプトをシステム構成の変更後に使用したりすると、データが壊れたり失われたりすることがあります。

▼ 保存された metassist コマンドのシェルスクリプトを実行する方法

始める前に [262 ページの「ボリュームをトップダウン作成するための前提条件」](#)を確認してください。

- 1 シェルスクリプトの作成後にシステム構成が変更されていないこと、さらに、スクリプトを実行するシステムがこのスクリプトを作成したシステムであることを確認します。
- 2 保存されたシェルスクリプトを実行します。

```
# sh ./metassist-shell-script-name
```

- 3 新しいボリュームを表示します。

```
# metastat -s diskset-name
```

例 23-8 保存された metassist コマンドのシェルスクリプトを実行する

次の例では、metassist コマンドでシェルスクリプトを使用してボリュームを作成する方法を示します。

```
# sh ./tmp/metassist-shell-script.sh
myset/hsp000: Hotspare pool is setup
myset/hsp000: Hotspare is added
myset/d2: Concat/Stripe is setup
myset/d1: Concat/Stripe is setup
myset/d0: Mirror is setup
myset/d0: submirror myset/d1 is attached
```

metassist コマンドによるボリューム構成ファイルの作成

metassist コマンドに `-d` 引数を指定すると、XML ベースのボリューム構成ファイルが生成されます。このファイルには、ボリュームに関連するすべてのオプションや情報など、ボリュームとそのコンポーネントの詳細が含まれています。このファイルを調べることによって、metassist コマンドが推奨する構成を知ることができます。さらに、このボリューム構成ファイルを慎重に変更して構成を微調整したあと、実際のボリューム作成の際に metassist コマンドへの入力として使用することもできます。

▼ metassist コマンドを使用してボリューム構成ファイルを作成する方法

始める前に [262 ページの「ボリュームをトップダウン作成するための前提条件」](#)を確認してください。

- 1 ボリュームの作成に使用するストレージを特定します。
ストレージを明示的に指定しなかった場合、システム上の未使用のストレージを Solaris Volume Manager が特定し、必要に応じて使用します。ストレージを指定する場合は、広義(たとえば、コントローラ 1 のすべてのストレージ)または狭義(たとえば、c1t4d2 は使用し、c1t4d1 は使用しない)に指定でき、そのストレージが Solaris Volume Manager で使用されます。

2 タスクに応じて、**metassist** コマンドと適切なオプションを使用します。

-d オプションを使用して、ボリュームを実際には作成しないことを指定します。代わりに、XML ベースのボリューム構成ファイルが標準出力に送信されます。

```
# metassist create -s diskset-name -f -r redundancy -a device1, device2... \
  -S size -v verbosity [-d]
```

create	ボリュームを作成するために使用するサブコマンドです。
-s diskset-name	ボリュームに使用するディスクセットの名前を指定します。
-f	ボリュームとホットスペアを対応付けることを指定します。
-r redundancy	作成する冗長レベル (データコピー数) を指定します。
-a device1, device2...	ボリュームの作成に使用できるデバイスを指定します。
-S size	作成するボリュームのサイズを KB (キロバイト)、MB (メガバイト)、GB (ギガバイト)、または TB (テラバイト) 単位で指定します。
-v verbosity	出力の詳細度を指定します。指定できる値の範囲は 0 (出力がほとんどない) から 2 (出力が多い) です。デフォルトレベルは 1 (中程度の出力) です。
-d	ボリュームを実際には作成しないことを指定します。

注 **-d** 引数によって要求された XML ベースのボリューム構成ファイルは標準出力に送信されます。ただし、**metassist** コマンドの他の出力は標準エラーに送信されます。出力ストリームは自由にリダイレクトできます。

詳細は、次の例と **metassist(1M)** のマニュアルページを参照してください。

例 23-9 metassist コマンドを使用してボリューム構成ファイルを作成する

この例では、**metassist** コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。**-f** オプションで障害耐性を指定します。最後の引数 (**-d**) で、ボリュームを実際には作成しないことを指定します。代わりに、指定した構成を最終的に作成するためのボリューム構成ファイルが、標準出力に送信されます。

```
# metassist create -s myset -f -r 2 -S 10mb -d
```

```
.(output truncated)
```

```

Volume request completed successfully.
<?xml version="1.0"?>
<!DOCTYPE volume-config SYSTEM "/usr/share/lib/xml/dtd/volume-config.dtd">
  <volume-config>
    <diskset name="myset"/>
    <disk name="c1t3d0"/>
    <disk name="c1t6d0"/>
    <disk name="c1t4d0"/>
    <slice name="c1t3d0s7" sizeinblocks="0"/>
    <slice name="c1t3d0s0" sizeinblocks="17649765" startsector="10773"/>
    <slice name="c1t6d0s7" sizeinblocks="0"/>
    <slice name="c1t6d0s0" sizeinblocks="17649765" startsector="10773"/>
    <slice name="c1t4d0s7" sizeinblocks="0"/>
    <slice name="c1t4d0s0" sizeinblocks="17649765" startsector="10773"/>
    <hsp name="hsp000">
    <slice name="c1t3d0s1" sizeinblocks="21546" startsector="17660538"/>
    </hsp>
    <mirror name="d0" read="ROUNDROBIN" write="PARALLEL" passnum="1">
    <concat name="d2">
    <slice name="c1t4d0s1" sizeinblocks="21546" startsector="17660538"/>
    <hsp name="hsp000"/>
    </concat>
    <concat name="d1">
    <slice name="c1t6d0s1" sizeinblocks="21546" startsector="17660538"/>
    <hsp name="hsp000"/>
    </concat>
    </mirror>
  </volume-config>
#

```

例 23-10 metassist コマンドを使用してボリューム構成ファイルを保存する

この例では、metassist コマンドを使用して、容量が 10M バイトの 2 面ミラーとホットスペアを作成し、障害耐性を強化する方法を示します。-f オプションで障害耐性を指定します。最後の引数 (-d) で、ボリュームを実際には作成しないことを指定します。代わりに、指定した構成を最終的に作成するためのボリューム構成ファイルが、標準出力に送信されます。コマンドの最後で、標準出力をリダイレクトして /tmp/metassist-volume-config.xml ボリューム構成ファイルを作成します。これはあとで、指定したボリュームを作成するために使用できます。

```
# metassist create -s myset -f -r 2 -S 10mb -d > \
/tmp/metassist-volume-config.xml
```

metassist コマンドのデフォルト動作の変更

ボリュームデフォルトファイル (/etc/defaults/metassist.xml) を使用して、metassist コマンドのデフォルト動作を変更できます。デフォルトファイルを変

更することによって、特定のディスクやコントローラを明示的に考慮から除外したり、考慮に含めたりできます。metassist コマンドで使用する大部分のボリューム設定値に要件を指定することもできます。

/etc/defaults/metassist.xml の形式は、文書型定義 (DTD) /usr/share/lib/xml/dtd/volume-defaults.dtd で規定されています。この形式については、[volume-defaults\(4\)](#) のマニュアルページを参照してください。

ボリュームデフォルトファイルの変更

ボリュームデフォルトファイル (/etc/defaults/metassist.xml) を編集して、metassist コマンドの動作を指定します。

注- ファイルを編集する際には、ファイルが文書型定義 (DTD) /usr/share/lib/xml/dtd/volume-defaults.dtd に準拠するようにしてください。XML ファイルが DTD に準拠していないと、metassist コマンドはエラーメッセージを出して異常終了します。

例 23-11 metassist コマンドを使ってデフォルト設定を変更したボリュームを作成する

ボリュームを作成する前に、/etc/default/metassist.xml ファイルを編集して、metassist コマンドで作成するすべてのボリュームに適用するデフォルト設定を指定します。この例に示す metassist コマンドでは、ボリュームをコントローラ c1 上にだけ作成します。さらに、ストライプを作成する際には必ず、4つのコンポーネントと飛び越し値 512KB からなるストライプを作成します。/etc/default/metassist.xml ファイルが再び変更されない限り、これらの制約はすべての metassist コマンドに適用されます。

```
# cat /etc/default/metassist.xml
<!DOCTYPE volume-defaults SYSTEM \
"/usr/share/lib/xml/dtd/volume-defaults.dtd">

<volume-defaults>
<available name="c1" />
<stripe mincomp="4" maxcomp="4" interlace="512KB" ></stripe>
</volume-defaults>

# metassist create -s myset -S 10Gb
```

この metassist コマンドによって、/etc/default/metassist.xml ファイルで指定されたとおりに、4つのスライスと 512K バイトの飛び越し値を使用して 10G バイトのストライプが作成されます。

モニタリングとエラー報告(タスク)

Solaris Volume Manager で、スライスレベルの物理エラーが原因でボリュームに書き込みできないというような問題が起きることがあります。問題が発生すると、Solaris Volume Manager がボリュームのステータスを変更するので、システム管理者は常に情報を把握できます。ただし、Solaris 管理コンソールを通じて Solaris Volume Manager の GUI を使用したり、`metastat` コマンドを実行してステータスを定期的にチェックしないと、ステータスの変化をタイムリーに把握することはできません。

この章では、Solaris Volume Manager SNMP エージェント (Solstice Enterprise Agents モニタリングソフトウェアのサブエージェント) など、Solaris Volume Manager のさまざまなモニタリングツールについて説明します。このツールを構成して SNMP トラップを報告する以外に、シェルスクリプトを作成して Solaris Volume Manager のさまざまな機能を能動的にモニターすることもできます。このシェルスクリプトは cron ジョブとして実行でき、潜在的な問題が顕在化する前にそれらを検出する上で役立ちます。

この章の内容は次のとおりです。

- 278 ページの「Solaris Volume Manager のモニタリングと報告(タスクマップ)」
- 278 ページの「エラーを周期的にチェックするための `mdmonitord` コマンドの構成」
- 279 ページの「Solaris Volume Manager SNMP エージェントの概要」
- 280 ページの「Solaris Volume Manager SNMP エージェントの構成」
- 282 ページの「Solaris Volume Manager SNMP エージェントの制限」
- 283 ページの「cron ジョブによる Solaris Volume Manager のモニタリング」

Solaris Volume Manager のモニタリングと報告 (タスクマップ)

次のタスクマップは、Solaris Volume Manager のエラー報告を管理するために必要な手順を示します。

タスク	説明	参照先
mdmonitord デーモンを構成してエラーを周期的にチェックする	/lib/svc/method/svc-mdmonitor スクリプトを編集して、mdmonitord デーモンに使用させるエラーチェック間隔を構成します。	278 ページの「エラーを周期的にチェックするための mdmonitord コマンドの構成」
Solaris Volume Manager SNMP エージェントを構成する	/etc/snmp/conf ディレクトリの構成ファイルを編集して、Solaris Volume Manager が正しいシステムにトラップを適切に送信できるようにします。	280 ページの「Solaris Volume Manager SNMP エージェントの構成」
cron コマンドでスクリプトを実行して Solaris Volume Manager をモニターする	エラーをチェックするスクリプトを作成または変更し、cron コマンドでスクリプトを実行します。	283 ページの「cron ジョブによる Solaris Volume Manager のモニタリング」

エラーを周期的にチェックするための mdmonitord コマンドの構成

Solaris Volume Manager には /usr/sbin/mdmonitord デーモンが組み込まれています。ディスクに障害が発生すると、Solaris Volume Manager は障害を検出し、エラーを生成します。このエラーイベントによって mdmonitord デーモンが起動され、RAID-1 (ミラー) ポリユーム、RAID-5 ポリユーム、およびホットスペアのチェックを実行します。ただし、指定の間隔でエラーを能動的にチェックするようにこのプログラムを構成することもできます。

▼ mdmonitord コマンドを構成してエラーを周期的にチェックする方法

/lib/svc/method/svc-mdmonitor スクリプトを編集して、周期的なチェックの時間間隔を追加します。

- 1 スーパーユーザーになります。

- 2 任意のエディタで `/lib/svc/method/svc-mdmonitor` スクリプトを開きます。スクリプト内で次のセクションを見つけます。

```
$MDMONITORD
error=$?
case $error in
0)      exit 0
        ;;

*)      echo "Could not start $MDMONITORD. Error $error."
        exit 0
```

- 3 `mdmonitord` コマンドから始まる行に、`-t` フラグとチェック間隔の秒数を追加します。

```
$MDMONITORD -t 3600
error=$?
case $error in
0)      exit 0
        ;;

*)      echo "Could not start $MDMONITORD. Error $error."
        exit 0
        ;;
esac
```

- 4 `mdmonitord` コマンドを再起動して変更を有効にします。

```
# svcadm restart system/mdmonitor
```

詳細は、[mdmonitord\(1M\)](#) のマニュアルページを参照してください。

Solaris Volume Manager SNMP エージェントの概要

Solaris Volume Manager SNMP トラップエージェントには、コアパッケージ `SUNWlvmr` と `SUNWlvma` のパッケージのほか、`Solstice Enterprise Agents` が必要です。これらのコアパッケージには、次のものがあります。

- `SUNWmibii`
- `SUNWsacom`
- `SUNWsadmi`
- `SUNWsasnm`

これらのパッケージは Solaris オペレーティングシステムの一部です。これらは、インストール時にパッケージ選択を変更した場合、または最小限のパッケージセットをインストールした場合を除き、通常はデフォルトでインストールされます。これらのパッケージが使用可能かどうかを確認するには、`pkginfo SUNWsasnm` のように、`pkginfo pkgname` コマンドを使用します。5つのパッケージがすべて使用できることを確認してから、次のセクションの説明に従って Solaris Volume Manager SNMP エージェントを構成する必要があります。

Solaris Volume Manager SNMP エージェントの構成

Solaris Volume Manager SNMP エージェントはデフォルトでは有効にされていません。SNMP トラップを有効にするには、次の手順に従います。

おそらく、Solaris 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、`/etc/snmp/conf/enterprises.oid` ファイルを編集し、その終わりに手順 6 の行を追加してから、Solaris Enterprise Agents サーバーの再起動を行う必要があります。

この手順を完了すると、指定したホストに SNMP トラップが送信されるようになります。送信されるトラップを表示するには、Solstice Enterprise Agents ソフトウェアなど、適切な SNMP モニターを使用する必要があります。

問題が発生したときにトラップを受信するためには、`mdmonitord` コマンドを設定してシステムを定期的にチェックする必要があります。278 ページの「エラーを定期的にチェックするための `mdmonitord` コマンドの構成」を参照してください。また、その他のエラーチェックオプションについては、283 ページの「cron ジョブによる Solaris Volume Manager のモニタリング」を参照してください。

▼ Solaris Volume Manager SNMP エージェントを構成する方法

- 1 スーパーユーザーになります。
- 2 `/etc/snmp/conf/mdlogd.rsrc` 構成ファイルを `/etc/snmp/conf/mdlogd.rsrc` に移動します。

```
# mv /etc/snmp/conf/mdlogd.rsrc- /etc/snmp/conf/mdlogd.rsrc
```
- 3 `/etc/snmp/conf/mdlogd.ac1` ファイルを編集して、SNMP トラップをどのホストに送信するかを指定します。ファイル内で次のものを探します。

```
trap = {
  {
    trap-community = SNMP-trap
    hosts = corsair
    {
      enterprise = "Solaris Volume Manager"
      trap-num = 1, 2, 3
    }
  }
}
```

`hosts = corsair` の行に、Solaris Volume Manager SNMP トラップの送信先ホストの名前を指定します。たとえば、SNMP トラップを `lexicon` に送信する場合は、この行を `hosts = lexicon` に変更します。複数のホストを指定する場合は、`hosts = lexicon, idiom` のように、ホスト名をコンマで区切って指定します。

- 4 また、`/etc/snmp/conf/snmpdx.acl` ファイルを編集して、**SNMP** トラップをどのホストに送信するかを指定します。

`trap =` で始まるブロックを探し、ここに前の手順で指定したものと同一ホスト名のリストを指定します。このセクションは `#` でコメント文にされていることがあります。その場合は、このセクションの中で必要な行の始めにある `#` を取り除いてください。トラップセクションにはコメントになっている行がほかにもあります。これらの行はそのまま残しておいても、わかりやすくするために削除してもかまいません。必要な行のコメントを解除し、`hosts` の行を変更した後のこのセクションは次のようになります。

```
#####
# trap parameters #
#####

trap = {
    {
        trap-community = SNMP-trap
        hosts =lexicon
        {
            enterprise = "sun"
            trap-num = 0, 1, 2-5, 6-16
        }
    }
    # {
    #     enterprise = "3Com"
    #     trap-num = 4
    # }
    # {
    #     enterprise = "snmp"
    #     trap-num = 0, 2, 5
    # }
    # }
    # {
    #     trap-community = jerry-trap
    #     hosts = jerry, nanak, hubble
    #     {
    #         enterprise = "sun"
    #         trap-num = 1, 3
    #     }
    #     {
    #         enterprise = "snmp"
    #         trap-num = 1-3
    #     }
    # }
}
```

注 - `/etc/snmp/conf/snmpdx.acl` ファイルに同じ数の左括弧と右括弧があることを確認してください。

- 5 前の手順でコメントを解除した、`/etc/snmp/conf/snmpdx.acl` ファイルのセクションの中に新しい **Solaris Volume Manager** セクションを追加します。

```
trap-community = SNMP-trap
hosts = lexicon
{
```

```
    enterprise = "sun"  
    trap-num = 0, 1, 2-5, 6-16  
  }  
  {  
    enterprise = "Solaris Volume Manager"  
    trap-num = 1, 2, 3  
  }  
}
```

追加された 4 行は、enterprise = "sun" ブロックのすぐ後に挿入されています。

- 6 /etc/snmp/conf/enterprises.oid ファイルの最後に次の行を追加します。

```
"Solaris Volume Manager" "1.3.6.1.4.1.42.104"
```

- 7 Solstice Enterprise Agents サーバーを停止し、再起動します。

```
# /etc/init.d/init.snmpdx stop  
# /etc/init.d/init.snmpdx start
```

Solaris Volume Manager SNMP エージェントの制限

Solaris Volume Manager SNMP エージェントは、システム管理者が把握すべき Solaris Volume Manager のあらゆる問題についてトラップを送信するわけではありません。具体的には、このエージェントは、次の場合にのみトラップを送信します。

- RAID-1 または RAID-5 のサブコンポーネントが「保守が必要」状態に移行した場合
- ホットスペアが稼働状態に切り替わった場合
- ホットスペアが再同期を開始した場合
- ホットスペアが再同期を完了した場合
- ミラーがオフラインにされた場合
- ディスクセットが別のホストによって取得されたため、現在のホストがパニック状態になった場合

RAID-0 ボリュームやソフトパーティションが置かれているディスクが使用不可になった場合のように、多くの問題では、デバイスに対する読み取りや書き込みが試行されたときでさえ SNMP トラップが生成されません。このような場合、SCSI または IDE エラーが一般的に報告されます。しかし、これらのエラーがモニタリングコンソールに報告されるようにするには、他の SNMP エージェントからトラップを発行する必要があります。

cron ジョブによる Solaris Volume Manager のモニタリング

▼ ボリュームのエラーを自動的にチェックする方法

- **Solaris Volume Manager** 構成のエラーを自動的にチェックするには、**cron** ユーティリティで定期的に行うことができるスクリプトを作成します。

次のスクリプト例は、必要に応じて変更することができます。

注 - このスクリプトは、Solaris Volume Manager のエラーチェックを自動化するための基本的なスクリプトです。各自の構成に合わせてこのスクリプトを変更する必要があります。

```
#
#!/bin/ksh
#ident "@(#)metacheck.sh 1.3 96/06/21 SMI"
# ident='%Z%M% %I% %E% SMI'
#
# Copyright (c) 1999 by Sun Microsystems, Inc.
#
# metacheck
#
# Check on the status of the metadevice configuration. If there is a problem
# return a non zero exit code. Depending on options, send email notification.
#
# -h
# help
# -s setname
# Specify the set to check. By default, the 'local' set will be checked.
# -m recipient [recipient...]
# Send email notification to the specified recipients. This
# must be the last argument. The notification shows up as a short
# email message with a subject of
# "Solaris Volume Manager Problem: metacheck.who.nodename.setname"
# which summarizes the problem(s) and tells how to obtain detailed
# information. The "setname" is from the -s option, "who" is from
# the -w option, and "nodename" is reported by uname(1).
# Email notification is further affected by the following options:
# -f to suppress additional messages after a problem
# has been found.
# -d to control the suppression.
# -w to identify who generated the email.
# -t to force email even when there is no problem.
# -w who
# indicate who is running the command. By default, this is the
# user-name as reported by id(1M). This is used when sending
# email notification (-m).
# -f
# Enable filtering. Filtering applies to email notification (-m).
```

```
# Filtering requires root permission. When sending email notification
# the file /etc/lvm/metacheck.setname.pending is used to
# controll the filter. The following matrix specifies the behavior
# of the filter:
#
#   problem_found   file_exists
#   yes             no           Create file, send notification
#   yes             yes          Resend notification if the current date
#                               (as specified by -d datefmt) is
#                               different than the file date.
#   no              yes          Delete file, send notification
#                               that the problem is resolved.
#   no              no           Send notification if -t specified.
#
# -d datefmt
# Specify the format of the date for filtering (-f). This option
# controls the how often re-notification via email occurs. If the
# current date according to the specified format (strftime(3C)) is
# identical to the date contained in the
# /etc/lvm/metacheck.setname.pending file then the message is
# suppressed. The default date format is "%D", which will send one
# re-notification per day.
# -t
# Test mode. Enable email generation even when there is no problem.
# Used for end-to-end verification of the mechanism and email addresses.
#
#
# These options are designed to allow integration of metacheck
# into crontab. For example, a root crontab entry of:
#
# 0,15,30,45 * * * * /usr/sbin/metacheck -f -w SVMcron \
# -d '\%D \%h' -m notice@example.com 2148357243.8333033@pager.example.com
#
# would check for problems every 15 minutes, and generate an email to
# notice@example.com (and send to an email pager service) every hour when
# there is a problem. Note the \ prior to the '%' characters for a
# crontab entry. Bounced email would come back to root@nodename.
# The subject line for email generated by the above line would be
# Solaris Volume Manager Problem: metacheck.SVMcron.nodename.local
#

# display a debug line to controlling terminal (works in pipes)
decho()
{
    if [ "$debug" = "yes" ] ; then
        echo "DEBUG: $" < /dev/null > /dev/tty 2>&1
    fi
}

# if string $1 is in $2-* then return $1, else return ""
strstr()
{
    typeset look="$1"
    typeset ret=""

    shift
# decho "strstr LOOK .$look. FIRST .$1."
    while [ $# -ne 0 ] ; do
        if [ "$look" = "$1" ] ; then
```

```

        ret="$look"
    fi
    shift
done
echo "$ret"
}

# if string $1 is in $2-* then delete it. return result
strdstr()
{
    typeset    look="$1"
    typeset    ret=""

    shift

#   decho "strdstr LOOK .$look. FIRST .$1."
    while [ $# -ne 0 ] ; do
        if [ "$look" != "$1" ] ; then
            ret="$ret $1"
        fi
    shift
done
echo "$ret"
}

merge_continued_lines()
{
    awk -e '\
BEGIN { line = "";} \
$NF == "\\\" { \
    $NF = ""; \
    line = line $0; \
    next; \
} \
$NF != "\\\" { \
    if ( line != "" ) { \
        print line $0; \
        line = ""; \
    } else { \
        print $0; \
    } \
}'
}

# trim out stuff not associated with metadevices
find_meta_devices()
{
    typeset    devices=""

#   decho "find_meta_devices .$*."
    while [ $# -ne 0 ] ; do
        case $1 in
            d+([0-9]) ) # metadevice name
                devices="$devices $1"
                ;;
        esac
    shift
done
echo "$devices"
}

```

```

# return the list of top level metadevices
toplevel()
{
    typeset    comp_meta_devices=""
    typeset    top_meta_devices=""
    typeset    devices=""
    typeset    device=""
    typeset    comp=""

    metastat$setarg -p | merge_continued_lines | while read line ; do
    echo "$line"
    devices='find_meta_devices $line'
    set -- $devices
    if [ $# -ne 0 ] ; then
        device=$1
        shift
        # check to see if device already referred to as component
        comp='strstr $device $comp_meta_devices'
        if [ -z $comp ] ; then
            top_meta_devices="$top_meta_devices $device"
        fi
        # add components to component list, remove from top list
        while [ $# -ne 0 ] ; do
            comp=$1
            comp_meta_devices="$comp_meta_devices $comp"
            top_meta_devices='strdstr $comp $top_meta_devices'
            shift
        done
    fi
done > /dev/null 2>&1
echo $top_meta_devices
}

#
# - MAIN
#
METAPATH=/usr/sbin
PATH=//usr/bin:$METAPATH
USAGE="usage: metacheck [-s setname] [-h] [[-t] [-f [-d datefmt]] \
    [-w who] -m recipient [recipient...]"

datefmt="%D"
debug="no"
filter="no"
mflag="no"
set="local"
setarg=""
testarg="no"
who='id | sed -e 's/^uid=[0-9][0-9]*(// -e 's/).*//''

while getopts d:Dfms:tw: flag
do
    case $flag in
    d)    datefmt=$OPTARG;
        ;;
    D)    debug="yes"
        ;;
    f)    filter="yes"
        ;;
    *)
    esac
done

```

```

;;
m)   mflag="yes"
;;
s)   set=$OPTARG;
if [ "$set" != "local" ] ; then
    setarg=" -s $set";
fi
;;
t)   testarg="yes";
;;
w)   who=$OPTARG;
;;
\?)  echo $USAGE
exit 1
;;
esac
done

# if mflag specified then everything else part of recipient
shift `expr $OPTIND - 1`
if [ $mflag = "no" ] ; then
    if [ $# -ne 0 ] ; then
        echo $USAGE
        exit 1
    fi
else
    if [ $# -eq 0 ] ; then
        echo $USAGE
        exit 1
    fi
fi
recipients="$*"

curdate_filter='date +%datefmt'
curdate='date'
node='uname -n'

# establish files
msg_f=/tmp/metacheck.msg.$$
msgs_f=/tmp/metacheck.msgs.$$
metastat_f=/tmp/metacheck.metastat.$$
metadb_f=/tmp/metacheck.metadb.$$
metahs_f=/tmp/metacheck.metahs.$$
pending_f=/etc/lvm/metacheck.$set.pending
files="$metastat_f $metadb_f $metahs_f $msg_f $msgs_f"

rm -f $files > /dev/null 2>&1
trap "rm -f $files > /dev/null 2>&1; exit 1" 1 2 3 15

# Check to see if metadb is capable of running
have_metadb="yes"
metadb$setarg > $metadb_f 2>&1
if [ $? -ne 0 ] ; then
    have_metadb="no"
fi
grep "there are no existing databases" < $metadb_f > /dev/null 2>&1
if [ $? -eq 0 ] ; then
    have_metadb="no"
fi

```

```

grep "/dev/md/admin" < $metadb_f > /dev/null 2>&1
if [ $? -eq 0 ] ; then
    have_metadb="no"
fi

# check for problems accessing metadbs
retval=0
if [ "$have_metadb" = "no" ] ; then
    retval=1
    echo "metacheck: metadb problem, can't run '$METAPATH/metadb$setarg'" \
        >> $msgs_f
else
    # snapshot the state
    metadb$setarg 2>&1 | sed -e '1d' | merge_continued_lines > $metadb_f
    metastat$setarg 2>&1 | merge_continued_lines > $metastat_f
    metahs$setarg -i 2>&1 | merge_continued_lines > $metahs_f

    #
    # Check replicas for problems, capital letters in the flags
    # indicate an error, fields are separated by tabs.
    #
    problem='awk < $metadb_f -F\t '{if ($1 ~ /[A-Z]/) print $1;}'
    if [ -n "$problem" ] ; then
        retval='expr $retval + 64'
        echo "\
metacheck: metadb problem, for more detail run:\n\t$METAPATH/metadb$setarg -i" \
            >> $msgs_f
    fi

    #
    # Check the metadvice state
    #
    problem='awk < $metastat_f -e \
        '/State:/ {if ($2 != "Okay" && $2 != "Resyncing") print $0;}'
    if [ -n "$problem" ] ; then
        retval='expr $retval + 128'
        echo "\
metacheck: metadvice problem, for more detail run:" \
            >> $msgs_f

    # refine the message to toplevel metadevices that have a problem
    top='toplevel'
    set -- $top
    while [ $# -ne 0 ] ; do
        device=$1
        problem='metastat $device | awk -e \
            '/State:/ {if ($2 != "Okay" && $2 != "Resyncing") print $0;}'
        if [ -n "$problem" ] ; then
            echo "\t$METAPATH/metastat$setarg $device" >> $msgs_f
            # find out what is mounted on the device
            mp='mount|awk -e '/\dev/md/dsk\/'$device'[ \t]/{print $1;}'
            if [ -n "$mp" ] ; then
                echo "\t\t$mp mounted on $device" >> $msgs_f
            fi
        fi
        shift
    done
fi

```



```

#
# Check the hotspares to see if any have been used.
#
problem=""
grep "no hotspare pools found" < $metahs_f > /dev/null 2>&1
if [ $? -ne 0 ] ; then
problem=awk < $metahs_f -e \
    '/blocks/ { if ( $2 != "Available" ) print $0;}'
fi
if [ -n "$problem" ] ; then
retval='expr $retval + 256'
echo "\
metacheck: hot spare in use, for more detail run:\n\t$METAPATH/metahs$setarg -i" \
    >> $msgs_f
fi

# If any errors occurred, then mail the report
if [ $retval -ne 0 ] ; then
if [ -n "$recipients" ] ; then
re=""
if [ -f $pending_f ] && [ "$filter" = "yes" ] ; then
re="Re: "
# we have a pending notification, check date to see if we resend
penddate_filter='cat $pending_f | head -1'
if [ "$curdate_filter" != "$penddate_filter" ] ; then
rm -f $pending_f > /dev/null 2>&1
else
if [ "$sdebug" = "yes" ] ; then
echo "metacheck: email problem notification still pending"
cat $pending_f
fi
fi
fi
if [ ! -f $pending_f ] ; then
if [ "$filter" = "yes" ] ; then
echo "$curdate_filter\n\tDate:$curdate\n\tTo:$recipients" \
    > $pending_f
fi
echo "\
Solaris Volume Manager: $node: metacheck$setarg: Report: $curdate" >> $msg_f
echo "\
-----" >> $msg_f
cat $msg_f $msgs_f | mailx -s \
    "${re}Solaris Volume Manager Problem: metacheck.$who.$set.$node" $recipients
fi
else
cat $msgs_f
fi
else
# no problems detected,
if [ -n "$recipients" ] ; then
# default is to not send any mail, or print anything.
echo "\
Solaris Volume Manager: $node: metacheck$setarg: Report: $curdate" >> $msg_f
echo "\
-----" >> $msg_f
if [ -f $pending_f ] && [ "$filter" = "yes" ] ; then
# pending filter exists, remove it and send OK

```

```
        rm -f $pending_f                > /dev/null 2>&1
        echo "Problem resolved"         >> $msg_f
        cat $msg_f | mailx -s \
        "Re: Solaris Volume Manager Problem: metacheck.$who.$node.$set" $recipients
    elif [ "$testarg" = "yes" ] ; then
        # for testing, send mail every time even though there is no problem
        echo "Messaging test, no problems detected" >> $msg_f
        cat $msg_f | mailx -s \
        "Solaris Volume Manager Problem: metacheck.$who.$node.$set" $recipients
    fi
else
    echo "metacheck: Okay"
fi
fi

rm -f $files                > /dev/null 2>&1
exit $retval
```

cron ユーティリティーを使ってスクリプトを起動する手順については、[cron\(1M\)](#) のマニュアルページを参照してください。

Solaris Volume Manager のトラブルシューティング (タスク)

この章では、Solaris Volume Manager に関連する問題をトラブルシューティングする方法について説明します。この章では、トラブルシューティングの一般的なガイドラインと、既知の問題を解決するための具体的な手順も示します。

この章の内容は次のとおりです。

- 292 ページの「Solaris Volume Manager のトラブルシューティング (タスクマップ)」
- 293 ページの「システムのトラブルシューティングの概要」
- 294 ページの「ディスクの交換」
- 296 ページの「ディスク移動の問題からの回復」
- 298 ページの「Solaris 10 リリースにアップグレードしたあとのデバイス ID の不一致」
- 300 ページの「ブートの問題からの回復」
- 307 ページの「状態データベース複製の障害からの回復」
- 310 ページの「ソフトパーティションの問題からの回復」
- 312 ページの「別のシステムからのストレージの回復」
- 319 ページの「ディスクセットの問題からの回復」
- 320 ページの「`ufsdump` コマンドによるマウント済みファイルシステムのバックアップの実行」
- 322 ページの「システム回復の実行」

この章では、Solaris Volume Manager のいくつかの問題とその適切な解決方法について説明します。この章ではすべてを扱うわけではなく、一般的なシナリオと回復手順を紹介します。

Solaris Volume Manager のトラブルシューティング (タスクマップ)

次のタスクマップは、Solaris Volume Manager のトラブルシューティングに必要な手順を示します。

タスク	説明	参照先
不良ディスクを交換する	ディスクを交換してから、新しいディスク上の状態データベースの複製と論理ボリュームを更新します。	294 ページの「不良ディスクを交換する方法」
ディスク移動の問題から回復する	ディスクを元の場所に戻すか、製品サポートに連絡します。	296 ページの「ディスク移動の問題からの回復」
/etc/vfstab 内の不適切なエントリを修正する	ミラーに対して fsck コマンドを実行してから、システムが正しくブートするように /etc/vfstab ファイルを編集します。	301 ページの「/etc/vfstab 内の不適切なエントリを修正する方法」
ブートデバイスの障害から回復する	ほかのサブミラーからブートします。	303 ページの「ブートデバイスの障害から回復する方法」
状態データベースの複製数の不足から回復する	metadb コマンドを使用して、使用不能な複製を削除します。	307 ページの「状態データベースの複製数の不足から回復する方法」
失われたソフトパーティションの構成データを回復する	metarecover コマンドを使用して、ソフトパーティションの構成データを回復します。	310 ページの「ソフトパーティションの構成データを回復する方法」
復旧済みディスクから Solaris Volume Manager 構成を回復する	新しいシステムにディスクを接続し、既存の状態データベースの複製から Solaris Volume Manager の構成を再構築します。	312 ページの「ローカルディスクセットからストレージを回復する方法」
別のシステムからストレージを回復する	既知のディスクセットから別のシステムへストレージをインポートします。	312 ページの「別のシステムからのストレージの回復」
アクセスできないディスクセットを削除する。	metaset コマンドを使用して、取得または使用できないディスクセットのレコードを削除します。	319 ページの「ディスクセットの問題からの回復」
Solaris Volume Manager ボリュームに格納されたシステム構成を回復する。	Solaris OS インストールメディアを使用して、Solaris Volume Manager ボリュームに格納されたシステム構成を回復します。	322 ページの「システム回復の実行」

システムのトラブルシューティングの概要

システムのトラブルシューティングの前提条件

Solaris Volume Manager に関連するストレージ管理の問題をトラブルシューティングするには、次のことが必要です。

- root 権限を持っている
- すべてのデータの最新バックアップを取っている

Solaris Volume Manager のトラブルシューティングの一般的なガイドライン

Solaris Volume Manager のトラブルシューティングを行うときは、次の情報を用意してください。

- `metadb` コマンドの出力
- `metastat` コマンドの出力
- `metastat -p` コマンドの出力
- `/etc/vfstab` ファイルのバックアップコピー
- `/etc/lvm/mddb.cf` ファイルのバックアップコピー
- `prtvtoc` コマンド (SPARC システム) または `fdisk` コマンド (x86 ベースのシステム) で出力されるディスクパーティションの情報
- システムの Solaris のバージョン
- Solaris のインストール済みパッチのリスト
- Solaris Volume Manager のインストール済みパッチのリスト

ヒント - Solaris Volume Manager 構成を更新したり、ストレージやオペレーティングシステムに関連するその他の変更をシステムに適用した場合は、その構成情報の最新コピーを生成してください。この情報は cron ジョブで自動的に生成することもできます。

一般的なトラブルシューティング方法

1つの手順で Solaris Volume Manager に関連するすべての問題を評価できるわけではありませんが、一般には次の手順が役立ちます。

1. 現在の構成に関する情報を収集します。

2. `metastat` や `metadb` コマンドの出力など、最新のステータス情報を調べます。この情報から、障害のあるコンポーネントがわかるはずです。
3. 障害が起こりそうなハードウェア部分をチェックします。
 - すべてが適切に接続されているか
 - 最近、停電がなかったか
 - 機器を変更または追加しなかったか

ディスクの交換

このセクションでは、Solaris Volume Manager 環境でのディスク交換の方法について説明します。



注意-不良ディスク上または不良ディスク上に構築したボリューム上でソフトウェアパーティションを使用していた場合は、新しいディスクを同じ物理位置に配置し、交換対象ディスクと同じ `c ntnd n` 番号を使用する必要があります。

▼ 不良ディスクを交換する方法

- 1 `/var/adm/messages` ファイルと `metastat` コマンドの出力を調べて、交換すべき不良ディスクを特定します。
- 2 不良ディスクに配置されている状態データベースの複製があればすべて特定します。

`metadb` コマンドを使用して複製を見つけます。

`metadb` コマンドで、不良ディスクに配置されている状態データベースの複製のエラーが報告されることがあります。この例では、`c0t1d0` のデバイスに問題があります。

```
# metadb
  flags      first blk      block count      /dev/dsk/c0t0d0s4
a m    u          16              1034
a      u          1050             1034
a      u          2084             1034
W pc lu0     16              1034
W pc lu0     1050            1034
W pc lu0     2084            1034
```

この出力から、ローカルディスク `c0t0d0` と `c0t1d0` の各スライス 4 に、状態データベースの複製が 3 つあることがわかります。`c0t1d0s4` スライスのフラグフィールドの `W` は、このデバイスに書き込みエラーがあることを示しています。`c0t0d0s4` スライスの 3 つの複製は正常です。

- 状態データベースの複製が配置されているスライスの名前と状態データベースの複製数を記録します。その後、状態データベースの複製を削除します。

状態データベースの複製の数は、`metadb` コマンド出力に同じスライス名が表示される行数と同じです。この例では、`c0t1d0s4` にある3つの状態データベースの複製を削除します。

```
# metadb -d c0t1d0s4
```



注意 - 不良の状態データベースの複製を削除すると複製の数が3以下になる場合は、**状態データベースの複製を追加**してから先に進みます。これによって、前と同じ構成情報が保たれます。

- 不良ディスクにホットスペアがないか探して、あれば削除します。

`metastat` コマンドを使用してホットスペアを見つけます。この例では、`c0t1d0s6` がホットスペアプール `hsp000` に含まれていたため、プールから削除します。

```
# metahs -d hsp000 c0t1d0s6
hsp000: Hotspare is deleted
```

- 不良ディスクを交換します。

この手順では、使用しているハードウェアと環境によって、`cfgadm` コマンド、`luxadm` コマンド、またはその他のコマンドを使用しなければならないことがあります。この手順の実行時には、使用しているハードウェアのドキュメントに記載されている手順に従って、Solaris 上でのこのディスクの状態を適切に操作してください。

- 新しいディスクのパーティションを再分割します。

`format` コマンドまたは `fmthard` コマンドを使用して、不良ディスクと同じスライス情報に基づいてディスクをパーティション分割します。不良ディスクに対する `prtvtoc` の出力がある場合は、`fmthard -s /tmp/failed-disk-prtvtoc-output` コマンドで新しいディスクをフォーマットできます。

- 状態データベースの複製を削除した場合は、同じ数の複製を適切なスライスに追加します。

この例では、`/dev/dsk/c0t1d0s4` を使用します。

```
# metadb -a -c 3 c0t1d0s4
```

- ディスク上のスライスが、RAID-5 ボリュームのコンポーネントである場合、あるいは RAID-0 ボリュームのコンポーネントで、それが RAID-1 ボリュームのサブミラーになっている場合は、各スライスに `metareplace -e` コマンドを実行します。

この例では、`/dev/dsk/c0t1d0s4` およびミラー `d10` を使用します。

```
# metareplace -e d10 c0t1d0s4
```

- 9 交換したディスクのスライス上にソフトパーティションが直接構築されている場合は、ソフトパーティションが含まれているスライスごとに `metarecover -m -p` コマンドを実行します。このコマンドによって、ディスク上にエクステントヘッダーが再作成されます。

この例では、`/dev/dsk/c0t1d0s4` でディスクのソフトパーティションのマーキングを再作成する必要があります。スライスをスキャンし、状態データベースの複製の情報に基づいてマーキングを再適用します。

```
# metarecover c0t1d0s4 -m -p
```

- 10 ディスク上のソフトパーティションが、RAID-5 ボリュームのコンポーネントである場合、あるいは RAID-0 ボリュームのコンポーネントで、それが RAID-1 ボリュームのサブミラーになっている場合は、各スライスに `metareplace -e` コマンドを実行します。

この例では、`/dev/dsk/c0t1d0s4` およびミラー `d10` を使用します。

```
# metareplace -e d10 c0t1d0s4
```

- 11 RAID-0 ボリューム上にソフトパーティションが構築されている場合は、各 RAID-0 ボリュームに `metarecover` コマンドを実行します。

この例では、RAID-0 ボリューム `d17` にソフトパーティションが構築されています。

```
# metarecover d17 -m -p
```

- 12 削除されたホットスペアを置き換え、それを 1 つまたは複数の適切なホットスペアプールに追加します。

この例では、ホットスペアプール `hsp000` に `c0t1d0s6` が含まれていました。このスライスをホットスペアプールに追加します。

```
# metahs -a hsp000 c0t1d0s6
hsp000: Hotspare is added
```

- 13 ソフトパーティションまたは冗長ボリュームが障害の影響を受けた場合は、バックアップからデータを復元します。冗長ボリュームだけが影響を受けた場合は、データを検証します。

すべてのボリュームのユーザーデータとアプリケーションデータを調べます。必要であれば、アプリケーションレベルの整合性チェックプログラムを実行するか、ほかの方法を使用して、データをチェックします。

ディスク移動の問題からの回復

このセクションでは、Solaris Volume Manager 環境でディスクを移動させたあとで発生する予想外の問題から回復する方法について説明します。

ディスク移動とデバイス ID の概要

Solaris Volume Manager では、特定のディスクと対応づけられたデバイス ID を使用して、Solaris Volume Manager 構成で使用されているあらゆるディスクを追跡します。ディスクを別のコントローラに移した場合、または SCSI ターゲット番号が変更された場合、通常は Solaris Volume Manager が移動を正しく認識して、関連するすべての Solaris Volume Manager レコードを相応に更新します。システム管理者の介入は不要です。まれに、Solaris Volume Manager がレコードを完全に更新できず、ブート時にエラーが報告されることがあります。

名前のないデバイスに関するエラーメッセージの解決

新しいハードウェアを追加したり、ハードウェアを移動させると (あるコントローラから別のコントローラに一連のディスクを移動させた場合など)、Solaris Volume Manager が、移動されたディスクに対応するデバイス ID を調べ、内部 Solaris Volume Manager レコードの *cnt ndn* 名を適切に更新します。レコードを更新できなかった場合は、`svc:/system/mdmonitor` サービスによって生成されたブートプロセスがブート時にコンソールに対してエラーを報告します。

```
Unable to resolve unnamed devices for volume management.
Please refer to the Solaris Volume Manager documentation,
Troubleshooting section, at http://docs.sun.com or from
your local copy.
```

この問題によってデータが消失したわけでも、特に何かが起きるわけでもありません。このエラーメッセージは、Solaris Volume Manager の名前レコードが部分的にしか更新されなかったことを意味します。`metastat` コマンドの出力に、前に使用されていた *cn tndn* 名の一部が示されます。この出力には、移動後の状態が反映された *cn tndn* 名の一部も示されます。

この条件下で Solaris Volume Manager 構成の更新が必要になった場合は、`meta*` コマンドを実行する際に、必ず `metastat` コマンドで報告された *cntnd n* 名を使用してください。

このエラー条件が発生した場合、次のいずれかを実行して条件を解消できます。

- すべてのディスクを元の場所に戻します。次に、再構成リブートを実行するか、(単一コマンドとして) 次のコマンドを実行します。

```
/usr/sbin/devfsadm && /usr/sbin/metadevadm -r
```

これらのコマンドが完了すると、エラー条件が解消されます。

- サポート担当者に連絡して指示を受けます。

注- このエラー条件はめったに発生しません。万一発生した場合は、高い確率で、ファイバチャネルに接続されたストレージが影響を受けます。

Solaris 10 リリースにアップグレードしたあとのデバイス ID の不一致

Solaris 10 リリースから、デバイス ID 出力の表示形式が新しくなりました。Solaris Volume Manager によるデバイス ID 出力の表示が新旧どちらの形式であるかは、デバイス ID 情報が状態データベースの複製に追加された時期によって決まります。

デバイス ID はこれまで 16 進値で表示されていました。新しい形式では、ASCII 文字列でデバイス ID が表示されます。多くの場合、次の例のように、変化はわずかです。

旧形式: `id1,ssd@w600c0ff00000000007ecd255a9336d00`

新形式: `id1,ssd@n600c0ff00000000007ecd255a9336d00`

次の例のように、変化が大きい場合もあります。

旧形式: `id1,sd@w4849544143484920444b3332454a2d33364e4320202020203433334239383939`

新形式: `id1,ssd@n600c0ff00000000007ecd255a9336d00`

Solaris 10 リリースにアップグレードした場合、以前の Solaris リリースで作成された既存のディスクセットに関連付けられているデバイス ID の形式は、Solaris Volume Manager 構成で更新されません。以前の Solaris リリースに戻さなければならなくなった場合、アップグレード後にディスクセットに対して行なった構成の変更は、以前のリリースでは使用できないことがあります。該当する構成変更は、次のとおりです。

- アップグレード以前に存在していたディスクセットへの新しいディスクの追加
- 新規ディスクセットの作成
- 状態データベースの複製の作成

このような構成変更は、ローカルディスクセットを含め、Solaris Volume Manager で作成可能なあらゆるディスクセットに影響を与える可能性があります。たとえば、Solaris 10 リリースで作成したディスクセットに対してこのような変更を実行した場合、そのディスクセットは以前の Solaris リリースにインポートできません。もう 1 つの例として、ミラー化されたルートの片側を Solaris 10 リリースにアップグレードしたあとで、ローカルディスクセットに対して構成変更を行うことがあります。その後、サブミラーを以前の Solaris リリースに戻した場合、このような変更は認識されません。

Solaris 10 OS の構成には、アップグレードの場合も含めて、常に新しい形式のデバイス ID が示されます。prtconf -v コマンドを使用すると、この情報を表示できます。一方、Solaris Volume Manager は、旧形式と新形式のどちらかを表示します。Solaris Volume Manager でどちらの形式が表示されるかは、ディスクを使い始めたときに稼働していた Solaris OS のバージョンによって決まります。Solaris Volume Manager が Solaris OS 構成のデバイス ID と形式は異なるが同じ内容を示しているかどうかを判断するには、metastat コマンドの出力と prtconf -v コマンドの出力を比較します。

次の例では、metastat コマンドの出力に表示される c1t6d0 のデバイス ID は、同じディスクに対する prtconf -v コマンドの出力に表示されるものと形式は異なりますが、内容は同じです。

```
# metastat
d127: Concat/Stripe
  Size: 17629184 blocks (8.4 GB)
  Stripe 0:
    Device      Start Block  Dbase  Reloc
    c1t6d0s2    32768        Yes    Yes

Device Relocation Information:
Device Reloc Device ID c1t6d0  Yes    id1,sd@w4849544143484920444b3332454a2d3336
4e43202020203433334239383939

# prtconf -v
.(output truncated)

.
.
sd, instance #6
  System properties:
    name='lun' type=int items=1
    value=00000000
    name='target' type=int items=1
    value=00000006
    name='class' type=string items=1
    value='scsi'
  Driver properties:
    name='pm-components' type=string items=3 dev=none
    value='NAME=spindle-motor' + '0=off' + '1=on'
    name='pm-hardware-state' type=string items=1 dev=none
    value='needs-suspend-resume'
    name='ddi-failfast-supported' type=boolean dev=none
    name='ddi-kernel-ioctl' type=boolean dev=none
  Hardware properties:
    name='devid' type=string items=1
    value='id1,@THITACHI_DK32EJ-36NC_____433B9899'

.
.
.(output truncated)
```

prtconf -v コマンドの出力に含まれている「instance #6」の行は、metastat コマンドの出力に含まれているディスク c1t6d0 と対応しています。prtconf -v コマンドの出力に含まれているデバイス ID id1,@THITACHI_DK32EJ-36NC_____433B9899 は、metastat コマンドの出力に含まれているデバイス ID id1,sd@w4849544143484920444b3332454a2d33364e4320202020203433334239383939 と対応しています。この出力の相違は、Solaris Volume Manager では metastat コマンドの出力に 16 進形式のデバイス ID が表示され、Solaris 10 OS 構成では prtconf コマンドの出力に ASCII 文字列が表示されることを示しています。

ブートの問題からの回復

Solaris Volume Manager ではルート (/)、swap、および /usr ディレクトリをミラー化できるので、システムのブート時に特殊な問題が発生することがあります。このような問題の原因は、ハードウェア障害またはオペレータエラーのどちらかです。このセクションでは、このような問題に対処するための手順について説明します。

次の表に、そのような問題の説明と適切な解決方法を示します。

表 25-1 Solaris Volume Manager での一般的なブートの問題

ブートの問題の原因	参照先
/etc/vfstab ファイルの情報が正しくありません。	301 ページの「 /etc/vfstab 内の不適切なエントリを修正する方法 」
十分な数の状態データベースの複製が定義されていません。	307 ページの「 状態データベースの複製数の不足から回復する方法 」
ブートデバイス(ディスク)に障害が発生しました。	303 ページの「 ブートデバイスの障害から回復する方法 」

ブートの問題の背景情報

- エラーのためにボリュームが Solaris Volume Manager によってオフラインにされた場合は、障害が発生したディスクにあるすべてのファイルシステムをアンマウントしてください。

個々のディスクスライスは独立しているため、同じディスクに複数のファイルシステムがマウントされていることがあります。ソフトウェアに障害が発生した場合には、同じディスクの他のスライスでもまもなく障害が発生するはずで、ディスクスライスに直接マウントされているファイルシステムは、Solaris Volume Manager によるエラー処理の保護の対象になりません。このようなファイルシステムをマウントしたままにしておくと、システムクラッシュによってデータを失う可能性があります。

- サブミラーを無効な状態やオフラインのままにしておく時間を最小限に抑えてください。再同期やオンラインバックアップの処理中は、ミラー化による保護は不完全になります。

/etc/vfstab 内の不適切なエントリを修正する方法

ルート (/) ファイルシステムをミラー化するときなどに、/etc/vfstab ファイルに不適切なエントリを作成した場合、システムは一見、正常にブートしているように見えます。その後、システムに障害が発生します。この状況を修正するためには、シングルユーザーモードで /etc/vfstab ファイルを編集する必要があります。

/etc/vfstab ファイル内の不適切なエントリを修正する大まかな手順は次のとおりです。

1. シングルユーザーモードでシステムをブートする
2. ミラーボリュームに対して fsck コマンドを実行する
3. 読み取り/書き込みオプションを有効にして、ファイルシステムを再マウントする
4. オプション: ルート (/) ミラーに対して metaroot コマンドを実行する
5. /etc/vfstab ファイルで、ファイルシステムのエントリがこのボリュームを正しく参照していることを確認する
6. システムをリブートする

▼ ルート (/) RAID-1 (ミラー) ボリュームを回復する

次の例では、ルート (/) ファイルシステムが 2 面ミラー d0 でミラー化されています。/etc/vfstab ファイルのルート (/) エントリは、何らかの理由でファイルシステムの元のスライスに戻っています。しかし、/etc/system ファイルの情報は、ミラー d0 からブートすることを示したままです。この状況は通常、metaroot コマンドを使用しないで /etc/system ファイルと /etc/vfstab ファイルを保守した場合に発生します。または、/etc/vfstab ファイルの古いコピーを現在の /etc/vfstab ファイルにコピーしたことが原因になる場合もあります。

不適切な /etc/vfstab ファイルの例を次に示します。

```
#device      device      mount      FS      fsck      mount      mount
#to mount    to fsck     point      type     pass     at boot   options
#
/dev/dsk/c0t3d0s0 /dev/rdisk/c0t3d0s0 /      ufs      1      no      -
/dev/dsk/c0t3d0s1 -            -        swap     -        no      -
/dev/dsk/c0t3d0s6 /dev/rdisk/c0t3d0s6 /usr    ufs      2      no      -
#
/proc        -            /proc    proc     -        no      -
swap        -            /tmp     tmpfs    -        yes     -
```

エラーがあるために、システムは、ブート時に自動的にシングルユーザーモードになります。

```
ok boot
...
configuring network interfaces: hme0.
Hostname: host1
mount: /dev/dsk/c0t3d0s0 is not this fstype.
setmnt: Cannot open /etc/mnttab for writing

INIT: Cannot create /var/adm/utmp or /var/adm/utmpx

INIT: failed write of utmpx entry:" "

INIT: failed write of utmpx entry:" "

INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): <root-password>
```

この時点で、ルート (/) ファイルシステムと /usr ファイルシステムは、読み取り専用としてマウントされています。次の手順に従ってください。

- 1 ルート (/) ミラーに対して **fsck** コマンドを実行します。

注-ルート (/) ミラーの正しいボリュームを使用するように注意してください。

```
# fsck /dev/md/rdisk/d0
** /dev/md/rdisk/d0
** Currently Mounted on /
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2274 files, 11815 used, 10302 free (158 frags, 1268 blocks,
0.7% fragmentation)
```

- 2 ルート (/) ファイルシステムを読み取り/書き込みファイルシステムとしてマウントし直し、**/etc/vfstab** ファイルを編集できるようにします。

```
# mount -o rw,remount /dev/md/dsk/d0 /
mount: warning: cannot lock temp file </etc/.mnt.lock>
```

- 3 **metaroot** コマンドを実行します。

```
# metaroot d0
```

このコマンドは、ルート (/) ファイルシステムが現在はボリューム **d0** 上にあることを示すように、**/etc/system** ファイルと **/etc/vfstab** ファイルを編集します。

- 4 `/etc/vfstab` ファイルに正しいボリュームのエントリが含まれていることを確認します。

`/etc/vfstab` ファイル内のルート (`/`) ファイルシステムのエントリは次のようになり、RAID-1 ボリュームを正しく参照しているはずでず。

```
#device          device          mount   FS    fsck   mount  mount
#to mount        to fsck         point   type  pass  at boot options
#
/dev/md/dsk/d0    /dev/md/rdsk/d0 /        ufs   1     no    -
/dev/dsk/c0t3d0s1 -                -       swap -     no    -
/dev/dsk/c0t3d0s6 /dev/rdsk/c0t3d0s6 /usr    ufs   2     no    -
#
/proc            -                /proc   proc  -     no    -
swap            -                /tmp    tmpfs -     yes   -
```

- 5 システムをリブートします。
システムは通常の動作に戻ります。

▼ ブートデバイスの障害から回復する方法

ルート (`/`) がミラー化されているシステムのブートデバイスに障害がある場合は、代替ブートデバイスを設定する必要があります。

このタスクの大まかな手順は次のとおりです。

- 代替ルート (`/`) サブミラーからシステムをブートする
- エラーのある状態データベースの複製とボリュームを特定する
- 不良ディスクを修復する
- 状態データベースの複製とボリュームを元の状態に復元する

ブートデバイスに障害が発生すると、最初に次のようなメッセージが表示されます。このメッセージは、アーキテクチャーによって異なる場合があります。

```
Rebooting with command:
Boot device: /iommu/sbus/dma@f,81000/esp@f,80000/sd@3,0
The selected SCSI device is not responding
Can't open boot device
...
```

このメッセージが表示された場合は、そのデバイスを書き留めておきます。その後、次の手順に従ってください。

- 1 ルート (`/`) の別のサブミラーからブートします。
この例では、6つの状態データベースの複製のうち2つだけがエラー状態なので、まだブートは可能です。そうでない場合は、アクセスできない状態データベースの複製をシングルユーザーモードで削除する必要があります。この手順については、[307 ページ](#)の「状態データベースの複製数の不足から回復する方法」を参照してください。

ルート (/) ファイルシステムのミラーを作成したときに、その手順の中で代替ブートデバイスを書き留めてあるはずですが、この例では、disk2 が代替ブートデバイスになります。

```
ok boot disk2
SunOS Release 5.9 Version s81_51 64-bit
Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.
Hostname: demo
...
demo console login: root
Password: <root-password>
Dec 16 12:22:09 host1 login: ROOT LOGIN /dev/console
Last login: Wed Dec 12 10:55:16 on console
Sun Microsystems Inc. SunOS 5.9 s81_51 May 2002
...
```

- 2 **metadb** コマンドを使用して、障害のある状態データベース複製の数を調べます。

```
# metadb
      flags          first blk   block count
M     p             unknown    unknown    /dev/dsk/c0t3d0s3
M     p             unknown    unknown    /dev/dsk/c0t3d0s3
a m   p   luo       16          1034      /dev/dsk/c0t2d0s3
a     p   luo       1050         1034      /dev/dsk/c0t2d0s3
a     p   luo       16          1034      /dev/dsk/c0t1d0s3
a     p   luo       1050         1034      /dev/dsk/c0t1d0s3
```

この例では、不良ディスクの一部であるスライス /dev/dsk/c0t3d0s3 上の状態データベースの複製は、システムから認識されなくなっています。

- 3 **metastat** コマンドを使用して、ルート (/)、**swap**、および **/usr** の各ミラーの半分に障害があることを確認します。

```
# metastat
d0: Mirror
   Submirror 0: d10
      State: Needs maintenance
   Submirror 1: d20
      State: Okay
...

d10: Submirror of d0
   State: Needs maintenance
   Invoke: "metareplace d0 /dev/dsk/c0t3d0s0 <new device>"
   Size: 47628 blocks
   Stripe 0:
   Device          Start Block  Dbase State          Hot Spare
   /dev/dsk/c0t3d0s0      0          No  Maintenance

d20: Submirror of d0
   State: Okay
   Size: 47628 blocks
   Stripe 0:
   Device          Start Block  Dbase State          Hot Spare
   /dev/dsk/c0t2d0s0      0          No  Okay

d1: Mirror
   Submirror 0: d11
```



```

    State: Needs maintenance
    Submirror 1: d21
      State: Okay
  ...

d11: Submirror of d1
  State: Needs maintenance
  Invoke: "metareplace d1 /dev/dsk/c0t3d0s1 <new device>"
  Size: 69660 blocks
  Stripe 0:
    Device          Start Block  Dbase State          Hot Spare
    /dev/dsk/c0t3d0s1      0      No  Maintenance
  ...

d21: Submirror of d1
  State: Okay
  Size: 69660 blocks
  Stripe 0:
    Device          Start Block  Dbase State          Hot Spare
    /dev/dsk/c0t2d0s1      0      No   Okay
  ...

d2: Mirror
  Submirror 0: d12
    State: Needs maintenance
  Submirror 1: d22
    State: Okay
  ...

d12: Submirror of d2
  State: Needs maintenance
  Invoke: "metareplace d2 /dev/dsk/c0t3d0s6 <new device>"
  Size: 286740 blocks
  Stripe 0:
    Device          Start Block  Dbase State          Hot Spare
    /dev/dsk/c0t3d0s6      0      No  Maintenance
  ...

d22: Submirror of d2
  State: Okay
  Size: 286740 blocks
  Stripe 0:
    Device          Start Block  Dbase State          Hot Spare
    /dev/dsk/c0t2d0s6      0      No   Okay
  ...

```

この例では、次のサブミラーの保守が必要であることが `metastat` コマンドで示されています。

- サブミラー d10、デバイス c0t3d0s0
 - サブミラー d11、デバイス c0t3d0s1
 - サブミラー d12、デバイス c0t3d0s6
- 4 システムを停止し、ディスクを交換します。 `format` コマンドまたは `fmthard` コマンドを使用して、障害の発生前と同じようにディスクをパーティション分割します。

ヒント-新しいディスクが既存ディスク(この例では、ミラーの変更のない側)とまったく同じ場合は、新しいディスクを簡単にフォーマットできます。それには、`prtvtoc /dev/rdisk/c0t2d0s2 | fmthard -s - /dev/rdisk/c0t3d0s2` コマンド(この例では `c0t3d0`)を使用します。

```
# halt
...
Halted
...
ok boot
...
# format /dev/rdisk/c0t3d0s0
```

5 システムをリブートします。

ルート (/) ミラーの他方の半分からリブートする必要があります。この代替ブートデバイスは、ミラーを作成したときに書き留めておいたものです。

```
# halt
...
ok boot disk2
```

6 `metadb` コマンドを使用して、障害のある状態データベース複製を削除してから、再び追加します。

```
# metadb
      flags          first blk   block count
M      p            unknown    unknown    /dev/dsk/c0t3d0s3
M      p            unknown    unknown    /dev/dsk/c0t3d0s3
a m    p luo        16         1034       /dev/dsk/c0t2d0s3
a      p luo        1050        1034       /dev/dsk/c0t2d0s3
a      p luo        16         1034       /dev/dsk/c0t1d0s3
a      p luo        1050        1034       /dev/dsk/c0t1d0s3
# metadb -d c0t3d0s3
# metadb -c 2 -a c0t3d0s3
# metadb
      flags          first blk   block count
a m    p luo        16         1034       /dev/dsk/c0t2d0s3
a      p luo        1050        1034       /dev/dsk/c0t2d0s3
a      p luo        16         1034       /dev/dsk/c0t1d0s3
a      p luo        1050        1034       /dev/dsk/c0t1d0s3
a      u            16         1034       /dev/dsk/c0t3d0s3
a      u            1050        1034       /dev/dsk/c0t3d0s3
```

7 `metareplace` コマンドを使用して、サブミラーを再び有効にします。

```
# metareplace -e d0 c0t3d0s0
Device /dev/dsk/c0t3d0s0 is enabled

# metareplace -e d1 c0t3d0s1
Device /dev/dsk/c0t3d0s1 is enabled

# metareplace -e d2 c0t3d0s6
Device /dev/dsk/c0t3d0s6 is enabled
```

しばらくすると、再同期が完了します。これで、また元のデバイスからブートできるようになります。

状態データベース複製の障害からの回復

ドライブ障害などのために、状態データベースの複製の定足数が満たされていないと、システムをマルチユーザーモードでリブートできなくなります。この状況は、Solaris Volume Managerが状態データベースの使用可能な複製が半数に満たないことを検出したときに、パニックに続いて発生することがあります。状態データベースの使用可能な複製が半数かそれより少ない場合にシステムをリブートしたときにも、この状況が発生することがあります。Solaris Volume Managerの用語では、状態データベースは「無効」状態になりました。この手順では、この問題から修復する方法について説明します。

▼ 状態データベースの複製数の不足から回復する方法

- 1 システムをブートします。
- 2 使用不能な状態データベースの複製を特定します。
`# metadb -i`
- 3 1つまたは複数のディスクが使用不能であることがわかっている場合は、それらのディスク上の状態データベースの複製を削除します。それ以外の場合は、障害のある状態データベースの複製(`metadb`の出力のステータスフラグ**W**、**M**、**D**、**F**、**R**で示される)を十分な数だけ削除して、状態データベースの複製の過半数が正常なものであるようにします。

```
# metadb -d disk-slice
```

ヒント-大文字のステータスフラグが設定されている状態データベースの複製は、エラー状態です。小文字のステータスフラグが設定されている状態データベースの複製は、正常に動作しています。

- 4 複製が削除されたことを確認します。
`# metadb`
- 5 システムをリブートします。

```
# reboot
```

- 6 必要であれば、ディスクを交換し、適切にフォーマットしてから、必要な状態データベースの複製をディスクに追加します。

74 ページの「状態データベースの複製の作成」の手順に従います。

交換用のディスクが用意できたら、システムを停止し、不良ディスクを交換してから、もう一度システムをリブートします。format コマンドまたは fmthard コマンドを使用して、障害の発生前と同じようにディスクをパーティション分割します。

例 25-1 状態データベースの複製の無効状態から回復する

次の例では、7つの状態データベースの複製を含むディスクに障害が発生したものとします。その結果、システムに与えられている正常な複製は3つだけです。システムパニックが発生し、マルチユーザーモードでリブートできなくなります。

```
panic[cpu0]/thread=70a41e00: md: state database problem

403238a8 md:mddb_commitrec_wrapper+6c (2, 1, 70a66ca0, 40323964, 70a66ca0, 3c)
  %l0-7: 0000000a 00000000 00000001 70bbcce0 70bbcd04 70995400 00000002 00000000
40323908 md:alloc_entry+c4 (70b00844, 1, 9, 0, 403239e4, ff00)
  %l0-7: 70b796a4 00000001 00000000 705064cc 70a66ca0 00000002 00000024 00000000
40323968 md:md_setdevname+2d4 (7003b988, 6, 0, 63, 70a71618, 10)
  %l0-7: 70a71620 00000000 705064cc 70b00844 00000010 00000000 00000000 00000000
403239f8 md:setnm_ioctl+134 (7003b968, 100003, 64, 0, 0, ffbffc00)
  %l0-7: 7003b988 00000000 70a71618 00000000 00000000 000225f0 00000000 00000000
40323a58 md:md_base_ioctl+9b4 (157ffff, 5605, ffbffa3c, 100003, 40323ba8, ff1b5470)
  %l0-7: ff3f2208 ff3f2138 ff3f26a0 00000000 00000000 00000064 ff1396e9 00000000
40323ad0 md:md_admin_ioctl+24 (157ffff, 5605, ffbffa3c, 100003, 40323ba8, 0)
  %l0-7: 00005605 ffbffa3c 00100003 0157ffff 0aa64245 00000000 7efefeff 81010100
40323b48 md:mdioctl+e4 (157ffff, 5605, ffbffa3c, 100003, 7016db60, 40323c7c)
  %l0-7: 0157ffff 00005605 ffbffa3c 00100003 0003ffff 70995598 70995570 0147c800
40323bb0 genunix:ioctl+1dc (3, 5605, ffbffa3c, ffffffff8, ffffffff0, ffbffa65)
  %l0-7: 0114c57c 70937428 ff3f26a0 00000000 00000001 ff3b10d4 0aa64245 00000000

panic:
stopped at      edd000d8:      ta      %icc,%g0 + 125
Type 'go' to resume

ok boot -s
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard
OpenBoot 3.11, 128 MB memory installed, Serial #9841776.
Ethernet address 8:0:20:96:2c:70, Host ID: 80962c70.

Rebooting with command: boot -s
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a File and args: -s
SunOS Release 5.9 Version s81_39 64-bit

Copyright 1983-2001 Sun Microsystems, Inc. All rights reserved.
configuring IPv4 interfaces: hme0.
Hostname: dodo
```

```

metainit: dodo: stale databases

Insufficient metadvice database replicas located.

Use metadb to delete databases which are broken.
Ignore any "Read-only file system" error messages.
Reboot the system when finished to reload the metadvice database.
After reboot, repair any broken database replicas which were deleted.

Type control-d to proceed with normal startup,
(or give root password for system maintenance): root-password
single-user privilege assigned to /dev/console.
Entering System Maintenance Mode

Jun  7 08:57:25 su: 'su root' succeeded for root on /dev/console
Sun Microsystems Inc.  SunOS 5.9      s81_39  May 2002
# metadb -i
      flags          first blk      block count
a m p l u          16             8192         /dev/dsk/c0t0d0s7
a   p l            8208            8192         /dev/dsk/c0t0d0s7
a   p l           16400           8192         /dev/dsk/c0t0d0s7
M   p              16             unknown      /dev/dsk/c1t1d0s0
M   p             8208            unknown      /dev/dsk/c1t1d0s0
M   p            16400           unknown      /dev/dsk/c1t1d0s0
M   p            24592           unknown      /dev/dsk/c1t1d0s0
M   p            32784           unknown      /dev/dsk/c1t1d0s0
M   p            40976           unknown      /dev/dsk/c1t1d0s0
M   p            49168           unknown      /dev/dsk/c1t1d0s0
# metadb -d c1t1d0s0
# metadb
      flags          first blk      block count
a m p l u          16             8192         /dev/dsk/c0t0d0s7
a   p l            8208            8192         /dev/dsk/c0t0d0s7
a   p l           16400           8192         /dev/dsk/c0t0d0s7
#

```

システムパニックが発生したのは、スライス `/dev/dsk/c1t1d0s0` 上で状態データベースの複製を検出できなくなったからです。このスライスは、不良ディスクに含まれているか、または障害のあるコントローラに接続されています。最初の `metadb -i` コマンドによって、このスライス上の複製のマスターブロックに問題があることがわかります。

無効状態の状態データベースの複製を削除する場合、ルート (`/`) ファイルシステムは読み取り専用になっています。 `mddb.cf` のエラーメッセージが表示されますが、無視してかまいません。

この時点で、システムは再び動作状態になりますが、状態データベースの複製は、本来の数より少なくなっているはずですが、障害が発生したストレージの一部を使用していたボリュームも、障害、エラー、またはホットスペア使用のいずれかになります。このような問題には速やかに対処するようにしてください。

ソフトパーティションの問題からの回復

このセクションでは、ソフトパーティションの構成情報を回復する方法について説明します。次の手順を使用するのは、状態データベースの複製がすべて失われている、なおかつ次のいずれもない場合に限定してください。

- `metastat -p` の出力の最新コピーまたは正確なコピー
- `md.cf` ファイルの最新コピーまたは正確なコピー
- 最新の `md.tab` ファイル

▼ ソフトパーティションの構成データを回復する方法

各ソフトパーティションの先頭部分には、ソフトパーティションエクステンツの開始位置を示すセクターがあります。このような隠れたセクターはエクステンツヘッダーと呼ばれます。このヘッダーはソフトパーティションのユーザーには見えません。Solaris Volume Manager の構成データがすべて失われた場合には、ディスクをスキャンして構成データの生成を試みることができます。

この手順は、失われたソフトパーティションの構成情報を回復する最後の手段です。`metarecover` コマンドを使用するのは、`metadb` ファイルと `md.cf` ファイルの両方が失われており、かつ `md.tab` ファイルも失われているか古くなっている場合に限定してください。

注- この手順が有効なのは、ソフトパーティション情報を回復する場合だけです。この手順で、その他の失われた構成を回復したり、他の Solaris Volume Manager ボリュームの構成情報を回復したりすることはできません。

注- ソフトパーティション上に構築された他の Solaris Volume Manager ボリュームが構成に含まれていた場合は、ソフトパーティションを回復してから他のボリュームを回復するようにしてください。

ソフトパーティションの構成情報は、デバイスと状態データベースに格納されています。どちらかのソースが破壊されている可能性があるため、どちらが信頼できるソースなのかを `metarecover` コマンドに知らせる必要があります。

最初に、`metarecover` コマンドを使用して、2つのソースが一致するかどうかを判定します。一致する場合は、`metarecover` コマンドを使用して変更を行うことはできません。ただし、`metarecover` コマンドから不一致が報告された場合は、出力を丹念に

調べ、ディスクが壊れているのか、それとも状態データベースが壊れているのかを判断する必要があります。さらに、`metarecover` コマンドを使用して、適切なソースに基づいて構成を再構築します。

- 1 **158** ページの「ソフトパーティションの構成のガイドライン」を読んでください。
- 2 `metarecover` コマンドを使用して、ソフトパーティションの回復情報を確認します。

```
# metarecover component -p -d
component    raw コンポーネントの cnt ndnsn 名を指定します
-p           ソフトパーティションを再作成することを指定します
-d           物理スライスをスキャンしてソフトパーティションのエクステン
            トヘッダーを検出することを指定します
```

例 25-2 ディスク上のエクステントヘッダーからソフトパーティションを回復する

```
# metarecover c1t1d0s1 -p -d
The following soft partitions were found and will be added to
your metadvice configuration.
  Name          Size      No. of Extents
  d10           10240      1
  d11           10240      1
  d12           10240      1
# metarecover c1t1d0s1 -p -d
The following soft partitions were found and will be added to
your metadvice configuration.
  Name          Size      No. of Extents
  d10           10240      1
  d11           10240      1
  d12           10240      1
WARNING: You are about to add one or more soft partition
metadevices to your metadvice configuration.  If there
appears to be an error in the soft partition(s) displayed
above, do NOT proceed with this recovery operation.
Are you sure you want to do this (yes/no)?yes
c1t1d0s1: Soft Partitions recovered from device.
bash-2.05# metastat
d10: Soft Partition
  Device: c1t1d0s1
  State: Okay
  Size: 10240 blocks
    Device          Start Block  Dbase Reloc
    c1t1d0s1         0            No    Yes
  Extent           Start Block          Block count
    0                1                    10240
d11: Soft Partition
  Device: c1t1d0s1
  State: Okay
```

```

Size: 10240 blocks
Device          Start Block  Dbase Reloc
c1t1d0s1        0           No      Yes

Extent          Start Block          Block count
0              10242                10240

d12: Soft Partition
Device: c1t1d0s1
State: Okay
Size: 10240 blocks
Device          Start Block  Dbase Reloc
c1t1d0s1        0           No      Yes

Extent          Start Block          Block count
0              20483                10240

```

この例では、すべての状態データベースの複製が誤って削除されているときに、ディスクから3つのソフトパーティションを回復します。

別のシステムからのストレージの回復

Solaris Volume Manager 構成を元のシステムから別のシステムに回復することもできます。

▼ ローカルディスクセットからストレージを回復する方法

システムの障害時には、ストレージを別のシステムに接続し、ローカルディスクセットから完全な構成を回復できます。たとえば、6つのディスクからなる外部ディスクバックを備えたシステムがあり、それらのディスクのいくつかに Solaris Volume Manager 構成と少なくとも1つの状態データベースの複製が置かれているとします。システムの障害時には、そのディスクバックを新しいシステムに物理的に移動して、新しいシステムがその構成を認識できるように設定します。この手順では、ディスクを別のシステムに移動して、ローカルディスクセットから構成を回復する方法を示します。

注 - この回復手順を使用できるのは、Solaris 9以降の Solaris Volume Manager ポリユームに限られます。

- 1 **Solaris Volume Manager** 構成が含まれている1つまたは複数のディスクを、**Solaris Volume Manager** 構成が存在していないシステムに接続します。

- 再構成リブートを実行して、新たに追加したディスクをシステムが認識できるようにします。

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

- 状態データベースの複製が含まれている(新たに追加したディスク上の)スライスのメジャー/マイナー番号を特定します。

ls -lL を使用し、グループ名と日付の間にある 2 つの数字を書き留めます。これらの数字がこのスライスのメジャー/マイナー番号です。

```
# ls -lL /dev/dsk/c1t9d0s7
brw-r----- 1 root sys 32, 71 Dec 5 10:05 /dev/dsk/c1t9d0s7
```

- 必要であれば、`/etc/name_to_major` 内でメジャー番号を検索して、メジャー番号に対応するメジャー名を特定します。

```
# grep " 32" /etc/name_to_major sd 32
```

- 有効な状態データベースの複製が新しいディスク上のどこにあるのかを **Solaris Volume Manager** に指示する情報を使用して、`/kernel/drv/md.conf` ファイルを更新します。

たとえば、`mddb_bootlist1` で始まる行にある `sd` を、手順 4 で特定したメジャー名で置き換えます。この例の `71` を、手順 3 で特定したマイナー番号で置き換えます。

```
#pragma ident "@(#)md.conf 2.2 04/04/02 SMI"
#
# Copyright 2004 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
# The parameters nmd and md_nsets are obsolete. The values for these
# parameters no longer have any meaning.
name="md" parent="pseudo" nmd=128 md_nsets=4;

# Begin MDD database info (do not edit)
mddb_bootlist1="sd:71:16:id0";
# End MDD database info (do not edit)
```

- リブートして、**Solaris Volume Manager** に構成を再ロードさせます。

次のようなメッセージがコンソールに表示されます。

```
volume management starting.
Dec 5 10:11:53 host1 metadevadm: Disk movement detected
Dec 5 10:11:53 host1 metadevadm: Updating device names in
Solaris Volume Manager
The system is ready.
```

- 構成を確認します。`metadb` コマンドを使用して状態データベースの複製のステータスを確認し、`metastat` コマンドを使用して各ボリュームのステータスを表示します。

```
# metadb
      flags          first blk      block count
a m p luo          16             8192          /dev/dsk/c1t9d0s7
```

```

a      luo      16      8192      /dev/dsk/c1t10d0s7
a      luo      16      8192      /dev/dsk/c1t11d0s7
a      luo      16      8192      /dev/dsk/c1t12d0s7
a      luo      16      8192      /dev/dsk/c1t13d0s7
# metastat
d12: RAID
    State: Okay
    Interlace: 32 blocks
    Size: 125685 blocks
Original device:
    Size: 128576 blocks
    Device      Start Block  Dbase State      Reloc  Hot Spare
    c1t11d0s3    330      No   Okay      Yes
    c1t12d0s3    330      No   Okay      Yes
    c1t13d0s3    330      No   Okay      Yes

d20: Soft Partition
    Device: d10
    State: Okay
    Size: 8192 blocks
    Extent      Start Block      Block count
    0           3592             8192

d21: Soft Partition
    Device: d10
    State: Okay
    Size: 8192 blocks
    Extent      Start Block      Block count
    0           11785            8192

d22: Soft Partition
    Device: d10
    State: Okay
    Size: 8192 blocks
    Extent      Start Block      Block count
    0           19978            8192

d10: Mirror
    Submirror 0: d0
    State: Okay
    Submirror 1: d1
    State: Okay
    Pass: 1
    Read option: roundrobin (default)
    Write option: parallel (default)
    Size: 82593 blocks

d0: Submirror of d10
    State: Okay
    Size: 118503 blocks
    Stripe 0: (interlace: 32 blocks)
    Device      Start Block  Dbase State      Reloc  Hot Spare
    c1t9d0s0    0           No   Okay      Yes
    c1t10d0s0   3591        No   Okay      Yes

d1: Submirror of d10
    State: Okay

```

```

Size: 82593 blocks
Stripe 0: (interlace: 32 blocks)
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t9d0s1        0           No  Okay        Yes
  c1t10d0s1       0           No  Okay        Yes

Device Relocation Information:
Device      Reloc  Device ID
c1t9d0     Yes   id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3487980000U00907AZ
c1t10d0    Yes   id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3397070000W0090A8Q
c1t11d0    Yes   id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3449660000U00904NZ
c1t12d0    Yes   id1,sd@SSEAGATE_ST39103LCSUN9.0GLS32655400007010H04J
c1t13d0    Yes   id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3461190000701001T
#
# metadb
      flags          first blk      block count
a m p  lu0          16             8192        /dev/dsk/c1t9d0s7
a      lu0          16             8192        /dev/dsk/c1t10d0s7
a      lu0          16             8192        /dev/dsk/c1t11d0s7
a      lu0          16             8192        /dev/dsk/c1t12d0s7
a      lu0          16             8192        /dev/dsk/c1t13d0s7
# metastat
d12: RAID
  State: Okay
  Interlace: 32 blocks
  Size: 125685 blocks
Original device:
  Size: 128576 blocks
  Device          Start Block  Dbase State      Reloc  Hot Spare
  c1t11d0s3       330         No  Okay        Yes
  c1t12d0s3       330         No  Okay        Yes
  c1t13d0s3       330         No  Okay        Yes

d20: Soft Partition
  Device: d10
  State: Okay
  Size: 8192 blocks
  Extent          Start Block      Block count
  0                3592             8192

d21: Soft Partition
  Device: d10
  State: Okay
  Size: 8192 blocks
  Extent          Start Block      Block count
  0                11785            8192

d22: Soft Partition
  Device: d10
  State: Okay
  Size: 8192 blocks
  Extent          Start Block      Block count
  0                19978            8192

d10: Mirror
  Submirror 0: d0
  State: Okay

```

```

Submirror 1: d1
  State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 82593 blocks

d0: Submirror of d10
  State: Okay
  Size: 118503 blocks
  Stripe 0: (interlace: 32 blocks)
    Device                Start Block  Dbase State      Reloc  Hot Spare
    c1t9d0s0                0           No  Okay       Yes
    c1t10d0s0              3591        No  Okay       Yes

d1: Submirror of d10
  State: Okay
  Size: 82593 blocks
  Stripe 0: (interlace: 32 blocks)
    Device                Start Block  Dbase State      Reloc  Hot Spare
    c1t9d0s1                0           No  Okay       Yes
    c1t10d0s1              0           No  Okay       Yes

Device Relocation Information:
Device      Reloc  Device ID
c1t9d0      Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3487980000U00907AZ1
c1t10d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3397070000W0090A8Q
c1t11d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3449660000U00904NZ
c1t12d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS32655400007010H04J
c1t13d0     Yes    id1,sd@SSEAGATE_ST39103LCSUN9.0GLS3461190000701001T0
# metastat -p
d12 -r c1t11d0s3 c1t12d0s3 c1t13d0s3 -k -i 32b
d20 -p d10 -o 3592 -b 8192
d21 -p d10 -o 11785 -b 8192
d22 -p d10 -o 19978 -b 8192
d10 -m d0 d1 1
d0 1 2 c1t9d0s0 c1t10d0s0 -i 32b
d1 1 2 c1t9d0s1 c1t10d0s1 -i 32b
#

```

既知のディスクセットからのストレージの回復

Solaris Volume Manager にディスクセットのデバイス ID サポートが組み込まれたことによって、既知のディスクセットからストレージを回復したり、ディスクセットを別のシステムにインポートしたりできるようになりました。metainport コマンドを使用すると、あるシステムから別のシステムに既知のディスクセットをインポートできます。両方のシステムにおいて、既存の Solaris Volume Manager 構成でデバイス ID がサポートされている必要があります。デバイス ID サポートの詳細については、[216 ページの「ディスクセット内の非同期共有ストレージ」](#)を参照してください。metainport コマンドの詳細については、[metainport\(1M\)](#)のマニュアルページを参照してください。

▼ インポートに使用可能なディスクセットに関するレポートを出力する方法

- 1 スーパーユーザーになります。
- 2 インポートに使用可能なディスクセットに関するレポートを取得します。

```
# metainport -r -v
-r システムへのインポートに使用可能な構成解除されたディスクセットのレ
  ポートを提供します。
-v 状態データベースの複製の場所に関する詳細な情報と、システム上においてイ
  ンポートに利用できる未構成のディスクセットのディスクのステータスに関す
  る詳細な情報を提供します。
```

例 25-3 インポートに使用可能なディスクセットに関する報告

次の例は、インポートに使用可能なディスクセットに関するレポートを出力する方法を示しています。

```
# metainport -r
Drives in regular diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
More info:
  metainport -r -v c1t2d0
Import: metainport -s <newsetname> c1t2d0
Drives in replicated diskset including disk c1t4d0:
  c1t4d0
  c1t5d0
More info:
  metainport -r -v c1t4d0
Import: metainport -s <newsetname> c1t4d0
```

```
# metainport -r -v c1t2d0
Import: metainport -s <newsetname> c1t2d0
Last update: Mon Dec 29 14:13:35 2003
Device      offset      length replica flags
c1t2d0       16          8192    a      u
c1t3d0       16          8192    a      u
c1t8d0       16          8192    a      u
```

▼ ディスクセットをあるシステムから別のシステムにインポートする方法

1 スーパーユーザーになります。

2 ディスクセットがインポート可能であることを確認します。

```
# metaimport -r -v
```

3 使用可能なディスクセットをインポートします。

```
# metaimport -s diskset-name drive-name
```

-s *diskset-name* 作成されるディスクセットの名前を指定します。

drive-name インポートされるディスクセットの状態データベースの複製が含まれているディスク (c#t#d#) を指定します。

4 ディスクセットがインポートされていることを確認します。

```
# metaset -s diskset-name
```

例 25-4 ディスクセットのインポート

次の例は、ディスクセットをインポートする方法を示しています。

```
# metaimport -s red c1t2d0
Drives in diskset including disk c1t2d0:
  c1t2d0
  c1t3d0
  c1t8d0
More info:  metaimport -r -v c1t2d0# metaset -s red
```

```
Set name = red, Set number = 1
```

Host	Owner
host1	Yes

Drive	Dbase
c1t2d0	Yes
c1t3d0	Yes
c1t8d0	Yes

ディスクセットの問題からの回復

次のセクションでは、ディスクセット関連の特定の問題から回復する方法の詳細を説明します。

ディスクセットの所有権を取得できないときには

システム障害、ディスク障害、通信リンクの障害などが原因で、どのノードからもディスクセットの所有権を取得できなくなり、ディスクセットのレコードを削除できない場合には、現在のホスト上にある Solaris Volume Manager 状態データベースの複製からディスクセットのレコードを削除することができます。

ディスクセットのレコードを削除しても、ディスクセットに含まれる状態データベースの情報には影響しないため、あとで `metainport` コマンドを使用してそのディスクセットをインポートできます (231 ページの「ディスクセットのインポート」を参照)。

Sun Cluster 構成からディスクセットを削除する必要がある場合は、次の手順を使用します。ただし、Sun Cluster 構成が存在しないときに使用する `-P` オプションではなく、`-c` オプションを使用します。

▼ ディスクセットを削除する方法

- 1 `metaset` コマンドを使用して、ディスクセットの取得を試みます。

```
# metaset -s setname -t -f
```

このコマンドは、`setname` で指定したディスクセットを強制的に (`-f`) に取得 (`-t`) します。ディスクセットを取得できた場合、このコマンドは成功します。このコマンドを実行したときに別のホストがこのディスクセットを所有していた場合、このホストはパニック状態になり、データの破損や損失が回避されます。このコマンドが成功した場合はディスクセットをきれいに削除でき、ディスクセットを削除する必要はありません。

セットを取得できない場合は、所有権レコードを削除することができます。

- 2 `metaset` コマンドに `-P` を付けて使用して、現在のホストからディスクセットを削除します。

```
# metaset -s setname -P
```

このコマンドは、コマンドを実行したホストから、`setname` で指定したディスクセットを削除 (`-P`) します。

- 3 **metaset** コマンドを使用して、セットが削除されたことを確認します。

```
# metaset
```

例 25-5 ディスクセットを削除する

```
host1# metaset -s red -t -f
metaset: host1: setname "red": no such set
```

```
host2# metaset
```

```
Set name = red, Set number = 1
```

```
Host          Owner
  host2
```

```
Drive  Dbase
```

```
c1t2d0  Yes
```

```
c1t3d0  Yes
```

```
c1t8d0  Yes
```

```
host2# metaset -s red -P
```

```
host2# metaset
```

- 参照
- ディスクセットに関連する概念については、第 18 章「ディスクセット (概要)」。
 - ディスクセットに関連するタスクについては、第 19 章「ディスクセット (タスク)」。

ufsdump コマンドによるマウント済みファイルシステムのバックアップの実行

次の手順では、RAID-1 ボリューム上のマウント済みファイルシステムを `ufsdump` コマンドでバックアップする場合に、コマンドのパフォーマンスを向上させる方法を示します。

▼ RAID-1 ボリューム上のマウント済みファイルシステムのバックアップを実行する方法

`ufsdump` コマンドを使用すると、RAID-1 ボリューム上のマウント済みファイルシステムのファイルをバックアップできます。バックアップユーティリティーが `ufsdump` の場合は、ボリュームの読み取りポリシーを「first」に設定します。これによってバックアップの実行速度が向上します。

- 1 スーパーユーザーになります。
- 2 **metastat** コマンドを実行して、ミラーが「正常」状態であることを確認します。

```
# metastat d40
d40: Mirror
  Submirror 0: d41
    State: Okay
  Submirror 1: d42
    State: Okay
  Pass: 1
  Read option: roundrobin (default)
  Write option: parallel (default)
  Size: 20484288 blocks (9.8 GB)
```

ミラーが「保守」状態の場合は、まずそれを修復する必要があります。

- 3 ミラーの読み取りポリシーを「**first**」に設定します。

```
# metaparam -r first d40
# metastat d40
d40: Mirror
  Submirror 0: d41
    State: Okay
  Submirror 1: d42
    State: Okay
  Pass: 1
  Read option: first
  Write option: parallel (default)
  Size: 20484288 blocks (9.8 GB)
```

- 4 ファイルシステムのバックアップを実行します。

```
# ufsdump 0f /dev/backup /opt/test
```

- 5 **ufsdump** コマンドの実行後、ミラーの読み取りポリシーを「**roundrobin**」に設定します。

```
# metaparam -r roundrobin d40
# metastat d40
d40: Mirror
  Submirror 0: d41
    State: Okay
  Submirror 1: d42
    State: Okay
  Pass: 1
  Read option: roundrobin
  Write option: parallel (default)
  Size: 20484288 blocks (9.8 GB)
```

システム回復の実行

システム回復を実行するには、DVDまたはCDメディア上にある Solaris OS インストールイメージからブートすると便利な場合があります。たとえば、`root` のパスワードをリセットする場合にも、インストールイメージを使用すると便利です。

Solaris Volume Manager 構成を使用している場合は、基になるディスクではなく、Solaris Volume Manager ポリリュームをマウントします。この手順は、ルート (/) ファイルシステムがミラー化されている場合には特に重要になります。Solaris Volume Manager は Solaris OS の一部であるため、Solaris Volume Manager ポリリュームをマウントすると、すべての変更がミラーの両側に反映されます。

Solaris OS DVD または CD-ROM のインストールイメージから Solaris Volume Manager ポリリュームにアクセスできるようにするには、次の手順を使用します。

▼ Solaris Volume Manager 構成を使用してシステムを回復する方法

Solaris OS インストール DVD または CD メディアからシステムをブートします。この手順は、Solaris ミニルートの `root` プロンプトから実行します。

- 1 **Solaris Volume Manager** 構成が含まれている基になるディスクを読み取り専用としてマウントします。

```
# mount -o ro /dev/dsk/c0t0d0s0 /a
```
- 2 `md.conf` ファイルを `/kernel/drv` ディレクトリにコピーします。

```
# cp /a/kernel/drv/md.conf /kernel/drv/md.conf
```
- 3 ファイルシステムをミニルートからアンマウントします。

```
# umount /a
```
- 4 この構成を読み込むように、**Solaris Volume Manager** ドライバを更新します。`update_drv` コマンドから出力される警告メッセージはすべて無視します。

```
# update_drv -f md
```
- 5 システムポリリュームを構成します。

```
# metainit -r
```
- 6 **Solaris Volume Manager** 構成に RAID-1 ポリリュームがある場合は、これらのポリリュームの再同期をとります。

```
# metasync mirror-name
```

- 7 **mount** コマンドを使用して **Solaris Volume Manager** ボリュームにアクセスできるはずで
す。

```
# mount /dev/md/dsk/volume-name /a
```

例 25-6 **Solaris Volume Manager** 構成を使用してシステムを回復する

```
# mount -o ro /dev/dsk/c0t0d0s0 /a
# cp /a/kernel/drv/md.conf /kernel/drv/md.conf
# umount /a
# update_drv -f md
Cannot unload module: md
Will be unloaded upon reboot.
Forcing update of md.conf.
devfsadm: mkdir failed for /dev 0xled: Read-only file system
devfsadm: inst_sync failed for /etc/path_to_inst.1359: Read-only file system
devfsadm: WARNING: failed to update /etc/path_to_inst
# metainit -r
# metasync d0
# mount /dev/md/dsk/d0 /a
```


重要な Solaris Volume Manager ファイル

この付録では、参照目的で、Solaris Volume Manager のファイルについて説明します。この付録は、次のセクションから構成されています。

- 325 ページの「システムファイルとスタートアップファイル」
- 326 ページの「手動で構成するファイル」

システムファイルとスタートアップファイル

このセクションでは、Solaris Volume Manager が正常に動作するのに必要なファイルについて説明します。いくつかの特別な構成の変更を除き、これらのファイルを使用したり、変更したりする必要はありません。

- /etc/lvm/mddb.cf



注意 - このファイルは編集しないでください。このファイルを変更すると、Solaris Volume Manager 構成を破壊することがあります。

/etc/lvm/mddb.cf ファイルには、状態データベースの複製の場所が記録されています。状態データベースの複製の場所が変更されると、Solaris Volume Manager は、すべての状態データベースの場所が記録されている mddb.cf ファイルにエントリを作成します。詳細については、[mddb.cf\(4\)](#) のマニュアルページを参照してください。

- /etc/lvm/md.cf

/etc/lvm/md.cf ファイルには、デフォルト (指定がないかローカルの) ディスクセットの自動的に生成された構成情報が格納されています。Solaris Volume Manager 構成が変更されると、Solaris Volume Manager は md.cf ファイルを自動的に更新します (使用中のホットスペアに関する情報を除く)。詳細については、[md.cf\(4\)](#) のマニュアルページを参照してください。



注意 - このファイルは編集しないでください。このファイルを変更すると、Solaris Volume Manager 構成が壊れたり、復元できなくなったりすることがあります。

状態データベースに保持されている情報が失われた場合でも、その間にボリュームが変更されたり、作成されたりしていない限り、`md.cf` ファイルを使って構成を復元できます。243 ページの「構成ファイルから Solaris Volume Manager を初期化する方法」を参照してください。

- `/kernel/drv/md.conf`

`md.conf` 構成ファイルは、Solaris Volume Manager によって起動時に読み取られます。`md.conf` ファイルには、状態データベースの複製の構成情報が格納されています。Solaris 10 から、`nmd` パラメータと `md_nsets` パラメータは手動で編集できなくなりました。Solaris Volume Manager が拡張され、必要に応じてボリュームが動的に構成されるようになりました。

手動で構成するファイル

md.tab ファイルの概要

`/etc/lvm/md.tab` ファイルには、Solaris Volume Manager の構成情報が格納されており、この情報を使って、Solaris Volume Manager 構成を再構築することができます。Solaris Volume Manager は、このファイルをコマンド行ユーティリティー `metainit`、`metadb`、`metabs` への入力として使用することによって構成を再構築できます。このファイルには、ボリューム、ディスクセット、ホットスペアプールのエントリが含まれる場合があります。このファイルを作成する手順 (`metastat -p > /etc/lvm/md.tab` コマンドを使用して) については、243 ページの「構成ファイルを作成する方法」を参照してください。

注 - `/etc/lvm/md.tab` ファイルの構成情報と、使用中の現在のボリューム、ホットスペア、状態データベースの複製が異なる場合があります。このファイルは、システム管理者が目的の構成を取得するために、手動で使用します。Solaris Volume Manager 構成を変更したあと、このファイルを再作成し、バックアップコピーを保管しておいてください。

このファイルを作成または更新したら、`metainit`、`metabs`、`metadb` コマンドで、このファイルに定義したボリューム、ホットスペアプール、状態データベースの複製をアクティブにします。

/etc/lvm/md.tab ファイルでは、各行に1つのボリュームの1つの完全な構成エントリが、`metainit`、`metadb`、および `metahs` コマンドの構文を使用して、表示されません。

注 - `metainit -an` コマンドを使って、`md.tab` ファイル内にあるすべてのボリュームの初期化をシミュレートすると、`md.tab` で定義されている他のボリュームへの依存関係を持つボリュームについてのエラーメッセージが表示されることがあります。こうしたエラーメッセージが表示されるのは、`metainit -an` の実行時に作成されたボリュームの状態を Solaris Volume Manager が維持していないためです。構成が存在している場合は、既存の構成に基づいて各行が評価されます。したがって、`metainit -an` コマンドが失敗するよう見えても、`-n` オプションを指定しなければ成功する可能性があります。

次に `metainit` コマンドを、`-a` オプションを指定して、`/etc/lvm/md.tab` ファイル内のすべてのボリュームをアクティブにするか、またはファイル内の特定のエントリに対応するボリューム名を指定して実行します。

注 - Solaris Volume Manager が `/etc/lvm/md.tab` ファイルに構成情報を書き込んだり、格納したりすることはありません。Solaris Volume Manager コンポーネントを作成するためには、このファイルを手動で編集し、`metainit`、`metahs`、または `metadb` コマンドを実行する必要があります。

詳細は、[md.tab\(4\)](#) のマニュアルページを参照してください。

Solaris Volume Manager のクイックリファレンス

この付録では、Solaris Volume Manager で使用可能な各機能に関するすぐにアクセス可能な情報を提供します。

コマンド行のリファレンス

次に、Solaris Volume Manager を管理するために使用するすべてのコマンドを示します。詳細は、各マニュアルページを参照してください。

表 B-1 Solaris Volume Manager コマンド

Solaris Volume Manager コマンド	説明	マニュアルページ
growfs	UFS ファイルシステムを非破壊的な方法で拡張します。	growfs(1M)
metaclear	アクティブボリュームとホットスペアプールを削除します。	metaclear(1M)
metadb	状態データベースの複製を作成および削除します。	metadb(1M)
metadetach	RAID-0 または RAID-1 (ミラー) ボリュームからボリュームを、またはトランザクションボリュームからログデバイスを切り離します。 注-トランザクションボリュームはサポートされなくなりました。	metadetach(1M)
metadevadm	デバイス ID 構成を確認します。	metadevadm(1M)
metahs	ホットスペアとホットスペアプールを管理します。	metahs(1M)

表 B-1 Solaris Volume Manager コマンド (続き)

Solaris Volume Manager コマンド	説明	マニュアルページ
metaimport	ディスクセット (複製されたディスクセットを含む) を、ディスクセットでデバイス ID がサポートされている既存の Solaris Volume Manager 構成にインポートします。	metaimport(1M)
metainit	ボリュームを構成します。	metainit(1m)
metaoffline	サブミラーをオフラインにします。	metaoffline(1M)
metaonline	サブミラーをオンラインにします。	metaonline(1M)
metaparam	ボリュームパラメータを変更します。	metaparam(1M)
metarecover	ソフトパーティションの構成情報を回復します。	metarecover(1M)
metarename	ボリューム名を変更および交換します。	metarename(1M)
metareplace	サブミラーや RAID-5 ボリューム内のコンポーネントを置き換えます。	metareplace(1M)
metaroot	ルート (/) ファイルシステムをミラー化するようにシステムファイルを設定します。	metaroot(1M)
metaset	ディスクセットを管理します。	metaset(1M)
metastat	ボリュームまたはホットスペアプールのステータスを表示します。	metastat(1M)
metasync	リポート中にボリュームを再同期します。	metasync(1M)
metattach	コンポーネントを RAID-0 または RAID-1 ボリュームに接続します。	metattach(1M)

Solaris Volume Manager CIM/WBEM API

Solaris Volume Manager の管理

Solaris Volume Manager CIM/WBEM アプリケーションプログラミングインタフェース (API) は、公開された標準ベースのプログラムインタフェースを提供し、Solaris Volume Manager の監視や構成を可能にします。この API は、Distributed Management Task Force (DMTF) の Common Information Model (CIM) に基づいています。DMTF の詳細については、<http://www.dmtf.org> を参照してください。

CIM は、「スキーマ」と呼ばれるデータモデルを定義し、スキーマは次を規定しています。

- Solaris Volume Manager デバイスの属性とそれに対する操作
- 各種 Solaris Volume Manager デバイス間の関係
- Solaris Volume Manager デバイスと、ファイルシステムなどのオペレーティングシステムのその他の側面との関係

このモデルは、Solaris Web Based Enterprise Management (WBEM) SDK を通じて使用できます。WBEM SDK は、CIM で規定されているシステム管理機能へのアクセスを可能にする Java テクノロジーに基づく API セットです。

CIM/WBEM SDK の詳細については、『[Solaris WBEM 開発ガイド](#)』を参照してください。

索引

A

auto-take ディスクセット, 205–206

C

cron コマンド, 283

D

DiskSuite ツール, 「グラフィカルインタフェース」を参照

E

/etc/lvm/md.cf ファイル, 325
/etc/lvm/mddb.cf ファイル, 325
/etc/vfstab ファイル, 150
不適切なエントリの修正, 301

F

fmthard コマンド, 305, 308
format コマンド, 305, 308

G

growfs コマンド, 246–248, 329
growfs コマンド, 46, 248

GUI, サンプル, 41

K

/kernel/drv/md.conf ファイル, 326

L

lockfs コマンド, 154

M

md.cf ファイル, 326
Solaris Volume Manager 構成の回復, 243
md.tab ファイル, 244
概要, 326–327
mdmonitord コマンド, 278–279
metaclear コマンド, 329
metaclear コマンド, 97, 147, 148, 150–152
metadb コマンド, 329
metadb コマンド, 77
metadb コマンド, 不良ディスクの交換と, 294
metadetach コマンド, 329
metadetach コマンド, 138, 147, 148
metadevadm コマンド, 329
不良ディスクの交換と, 294
metahs コマンド, 329
不良ディスクの交換と, 294
ホットスペアの置き換え, 200–202
ホットスペアの削除, 202–203

metahs コマンド (続き)
ホットスペアプールへのスライスの追加, 195-196
有効化, 203-204
metainport コマンド, 210-211, 231-233, 316-318, 330
metainit コマンド, 330
metainit コマンド, 245
metainit コマンド, ホットスペアプールの作成, 194-195
metaoffline コマンド, 330
metaoffline コマンド, 139
metaonline コマンド, 330
metaparam コマンド, 330
metaparam コマンド, 143
metaparam コマンド
ホットスペアプールの関連付けの変更, 198-199
ホットスペアプールのボリュームへの関連付け, 197-199
metarecover コマンド, 330
不良ディスクの交換と, 294
metarename コマンド, 241, 330
metarename コマンド, 242
metareplace コマンド, 330
metareplace コマンド, 140, 182, 306
metareplace コマンド, 不良ディスクの交換と, 294
metaroot コマンド, 330
metaset コマンド, 330
ディスクセットからのディスクの削除, 225-226
ディスクセットからのホストの削除, 229-231
ディスクセットの解放, 228-229
ディスクセットの削除, 229-231
ディスクセットの作成, 219-220
ディスクセットの取得, 226-228
ディスクセットのステータスの確認, 224-225
ディスクセットへのディスクの追加, 220
ディスクセットへのホストの追加, 222-223
metastat コマンド, 330
metastat コマンド, 142, 180
metasync コマンド, 330
metattach
タスク, 117, 122, 128, 133

metattach コマンド, 330
metattach コマンド, 96, 137, 144
RAID-5 コンポーネントの接続, 181
サブミラーの接続, 245

O

Oracle Real Application Clusters, 55-57

P

Solaris Volume Manager インタフェース
Solaris 管理コンソール, 41
コマンド行, 41
サンプル GUI, 41

Solaris Volume Manager 要素, 概要, 43

Solaris Volume Manager 構成の表示, 236-239

prtconf コマンド, デバイス ID の表示, 298-300

R

RAID, Solaris Volume Manager でサポートされるレベル, 32

RAID-0+1, 101-102

RAID-0 (ストライプ) ボリューム, 3つのスライスの例, 80

RAID-0 ボリューム

使用法, 79

定義, 79

RAID-1+0, 101-102

RAID-1 ボリューム, 99-102

RAID-0+1, 101-102

RAID-1+0, 101-102

オプション, 106-107

コンポーネントの置き換えと有効化, 248-252

コンポーネントの置き換えと有効化に関する情報, 251-252

作成, 113-135

サブミラー, 99-102

シングルユーザーモードでブートする, 109-110

パス番号, 107

- RAID-1 ボリューム (続き)
「保守」と「最後にエラー」の比較, 250-251
ミラー, 99-102
読み取り/書き込みポリシー, 106
- RAID-5 ボリューム
4つのディスクの例, 168
およびインターレース, 171
ガイドライン, 172
概要, 167-171
拡張, 180-181, 181
拡張デバイスの例, 169
コンポーネントの置き換えと有効化, 248-252
コンポーネントの置き換えと有効化に関する情報, 251
コンポーネントの交換と有効化, 174
コンポーネントの置換, 183-185
コンポーネントの有効化, 182
作成, 178-179
障害が発生したスライスの置換, 185
障害が発生したスライスの有効化, 182
状態, 172-174
ステータスのチェック, 179-180
スライスの再同期, 168
スライスの状態, 172-174
タスク, 177-178
定義, 32, 45
パリティ計算, 172
パリティ情報, 167, 170
「保守」と「最後にエラー」の比較, 250-251
要件, 171-172
- RAID-5 ボリューム内のスライスの有効化, 182
- RAID-0 (ストライプ) ボリューム, 80
作成するための情報, 87-88
- RAID-0 ボリューム
ガイドライン, 87-88
タイプ, 79
- RAID-0 (連結ストライプ) ボリューム
3つのストライプの例, 84
インターレース値, 84
- RAID-0 (連結) ボリューム, 作成するための情報, 87-88
- S**
- SCSI ディスク
交換, 294, 296
- SMF
Solaris Volume Manager サービス, 53-54
Solaris Volume Manager のアップグレード, 53-54
- Solaris Volume Manager
「Solaris Volume Manager」を参照
Oracle Real Applications Clusters, 55-57
Sun Cluster, 55-57
構成ガイドライン, 50
構成の回復, 243
ネットワーク接続されたストレージデバイス, 255
- Solaris Volume Manager for Sun Cluster
アプリケーションベースの回復, 60-62
構成, 60
最適化された再同期, 60-62
指定されたミラー読み取り, 60-62
ソフトウェアコンポーネント, 56-57
タイムアウト, 60
データの管理と回復, 60-62
- Sun Cluster, 55-57
- svc-mdmonitor スクリプト, 278-279
- swap, ミラー化の解除, 152
- V**
- /var/adm/messages ファイル, 248
不良ディスクの交換, 294
- あ**
アプリケーションベースの回復, 60-62
- い**
一般的なパフォーマンスに関するガイドライン, 35
インストールプログラム, システム回復, 322-323

インストールプログラムからのシステム回復, 322-323
インターレース値, 80
インタフェース, 「Solaris Volume Manager インタフェース」を参照
インタレース, 指定, 91

え

エラー, スクリプトを使用したチェック, 283-290
エラーチェック, 278-279

か

拡張ストレージ, 「グラフィカルインタフェース」を参照

き

共有ディスクセット, 205-206

く

グラフィカルインタフェース, 概要, 40

こ

構成, 表示, 236-239
構成の計画
 ガイドライン, 33
 概要, 33
 トレードオフ, 35
構成ファイル, 作成, 243-245
構成ファイルの作成, 243-245
コンポーネント, 定義, 79

さ

サービス管理機能 (SMF), 「SMF」を参照

再同期

完全, 103
最適化された, 103
部分的, 103
サブミラー, 100
 「RAID-1 ボリューム」を参照
 オフライン時の操作, 100
 障害が発生したスライスの置換, 145
 障害が発生したスライスの有効化, 140
 接続, 100
 切断, 100
 全体の置換, 147
サブミラー内のスライスの有効化, 140

し

ジオメトリック読み取りポリシー, 106
システムファイル, 325-326
指定されたミラー読み取り, 60-62
自動ディスクパーティション分割, 211-213
順次書き込みポリシー, 106
順次入出力, 37
小規模なサーバー, 配備の概要, 253-255
状態データベース
 概念的な概要, 48-49
 概念の概要, 68
 定義, 44, 48
 破損, 68
状態データベースの複製, 48
 2 ディスク構成, 71-72
 エラー, 71
 基本操作, 67
 最小数, 69-71
 サイズ, 211-213
 使用法, 67
 単一のスライスに複数を作成, 69-71
 定義, 48
 ディスクセットの条件, 211-213
 場所, 49, 69-71
 より大きい複製の追加, 77
シングルユーザーモードでブートする, 109-110

す

- ステータス, 225
- ストライプ
 - 拡張, 96
 - 削除, 97
 - 作成, 91
 - 定義, 80
- 「RAID-0(ストライプ)ボリューム」も参照
- ストライプボリューム, 「RAID-0(ストライプ)ボリューム」を参照
- スライス, 拡張, 95

せ

- 先頭読み取りポリシー, 106

そ

- ソフトパーティション, 157-158
 - ガイドライン, 158
 - 拡張, 164-165
 - 構成の回復, 310
 - 削除, 165-166
 - 作成, 162-163
 - ステータスのチェック, 163-166
 - タスク, 161
 - 定義, 157
 - 場所, 157
 - 保守, 163-166

た

- 多数決アルゴリズム, 67
- 単純なボリューム, 定義, 45

て

- ディスクセット, 205-206
 - 2つの共有ディスクセットのある例, 214
 - auto-take, 205-206
 - インポート, 210-211, 231-233

ディスクセット(続き)

- ガイドライン, 215
- 解放, 210, 228-229
- 管理, 208-214
- 共有, 205-206
- コンポーネントの作成, 223-224
- 作成, 219-220
- サポートされる最大数, 245
- 自動ディスクパーティション分割, 211-213
- シナリオ, 216
- 取得, 226-228
- 使用法, 205-206
- ステータスの確認, 224-225
- 定義, 44, 49-50
- ディスクセットからのホストの削除, 229-231
- ディスクセットの削除, 229-231
- ディスクの削除, 225-226
- ディスクの追加, 220
- 名前付き, 205-206
- 非同期共有ストレージ, 216
- 複数所有者, 205-206
 - 「複数所有者ディスクセット」を参照
- ホストの追加, 222-223
- 命名規則, 213-214
- 予約, 209-210, 227
- ローカル, 205-206
- ディスクセットからのディスクの削除, 225-226
- ディスクセットからのホストの削除, 229-231
- ディスクセット内でのコンポーネントの作成, 223-224
- ディスクセットのインポート, 210-211
- ディスクセットの解放, 210, 228-229
- ディスクセットの管理, 208-214
- ディスクセットの削除, 229-231
- ディスクセットの作成, 219-220
- ディスクセットの取得, 226-228
- ディスクセットのステータスの確認, 224-225
- ディスクセットの予約, 209-210
- ディスクセットへのディスクの追加, 220
- ディスクセットへのホストの追加, 222-223
- ディスクの再分割, 211-213
- デバイス ID, 形式, 298-300
- デフォルト値の変更, 245

- と
- トラブルシューティング
 - /etc/vfstab ファイルの不適切なエントリ, 301
 - metainport コマンド, 316-318
 - 一般的なガイドライン, 293
 - ディスク移動の問題からの回復, 296
 - ディスクセットのインポート, 316-318
 - デバイス ID の不一致, 298-300
 - ブートの問題, 300-307
 - 不良ディスクの交換, 294
- な
- 名前付きディスクセット, 205-206
- に
- 入出力, 36
- は
- パス番号
 - 定義済み, 107
 - 読み取り専用ミラー, 103
- ふ
- ファイルシステム
 - ガイドライン, 51
 - 拡張, 246-248
 - 拡張の概要, 46, 47
 - ミラー化の解除, 152
 - 連結を作成することによる拡張, 94-95
 - ブートの問題, 300-307
 - フェイルオーバー構成, 50
 - 複数所有者ディスクセット, 205-206
 - RAID-1 ボリューム, 60-62
 - インポート, 55-57
 - タスク, 58-59
 - 定義, 55-57
 - デバイス ID サポート, 55-57
 - 複数所有者ディスクセット (続き)
 - マスターノード, 57-59
 - 複製, 48
- へ
- 並列アクセス, 定義, 80
 - 並列書き込みポリシー, 106
- ほ
- ホットスペア
 - Solaris Volume Manager でのホットスペアのしくみ, 189
 - 概念の概要, 188
 - 定義, 188
 - ホットスペアプール内の置き換え, 202
 - ホットスペアプールへの追加, 196
 - 有効化, 203-204
 - ホットスペアの追加, 196
 - ホットスペアプール, 49
 - 概念の概要, 187-190
 - 関連付け, 198
 - 関連付けの変更, 198-199, 199
 - 基本操作, 49
 - 作成, 194-195
 - 状態, 189-190
 - ステータスの確認, 200-204
 - スライスの追加, 195-196
 - 定義, 44, 49, 188-189
 - 保守, 200-204
 - ホットスペアの置き換え, 200-202
 - ホットスペアの削除, 202-203
 - ボリュームへの関連付け, 197-199
 - ミラーの例, 190
 - ボリューム
 - 概念的な概要, 44
 - 仮想ディスク, 40
 - サポートされる最大数, 245
 - 使用, 45-46
 - タイプ, 45
 - 定義, 44
 - ディスク容量の拡張, 46-47

ボリューム (続き)
 名前の交換, 241
 名前の変更, 242
 ファイルシステムコマンドの使用, 46
 ボリュームのトップダウン作成
 RAID1 ボリューム, 作成, 264-267
 シェルスクリプト, 267-274
 デフォルト, 274-275
 ボリューム構成ファイル, 272
 ボリュームの名前の変更, 240
 ボリューム名の切り替え, 48
 ボリューム名の交換, 241

ま

マスターノード, 57-59

み

ミラー

「RAID-1 ボリューム」を参照
 「RAID-1 ボリューム」も参照
 2つのサブミラーの例, 100
 2面ミラー, 265-266, 266, 268-271, 271, 273-274,
 274, 275
 オプションの変更, 143
 オンラインバックアップ, 153
 拡張, 144
 コンポーネントの交換と有効化, 174
 サイズの更新, 143-144
 再同期, 102, 103
 作成, 113-135
 サブミラーの接続, 137
 ステータスの出力例, 141
 定義, 45

ミラー化

アンマウント可能なファイルシステム, 118
 ファイルシステム, 115-120
 読み取りおよび書き込みパフォーマンス, 34
 ルート (/), /usr、および swap, 119
 ルートファイルシステム
 GRUB の使用, 124-135
 SPARC, 120-124

ミラー化, ルートファイルシステム (続き)
 x86, 124-135

め

メタデバイス, 「ボリューム」を参照

ら

ラウンドロビン読み取りポリシー, 106
 ランダム入出力, 36-37

れ

連結

3つのスライスの例, 83
 UFS ファイルシステムの拡張, 82
 拡張, 96
 削除, 97
 作成, 93
 使用法, 82
 定義
 「RAID-0 (連結) ボリューム」も参照
 連結ストライプ
 「RAID-0 (連結ストライプ) ボリューム」を参
 照
 削除, 97
 定義
 「RAID-0 (連結ストライプ) ボリューム」も
 参照
 連結ボリューム, 「RAID-0 (連結) ボリューム」を
 参照

ろ

ローカルディスクセット, 205-206

