Veritas InfoScale™ 7.4.2 Oracle データベース用ストレージと可用性管理 - AIX、Linux、Solaris

VERITAS™

最終更新: 2020-07-15

法的通知

Copyright © 2020 Veritas Technologies LLC. All rights reserved.

Veritas および Veritas ロゴは、Veritas Technologies LLC または同社の米国およびその他の国における関連会社の商標または登録商標です。その他の会社名、製品名は各社の登録商標または商標です。

この製品には、サードパーティへの著作権を示す必要のあるサードパーティのソフトウェアが含まれる場合があります(「サードパーティプログラム」)。一部のサードパーティプログラムは、オープンソースまたはフリーソフトウェアライセンスの下で利用できます。本ソフトウェアに含まれる本使用許諾契約は、オープンソースまたはフリーソフトウェアライセンスでお客様が有する権利または義務を変更しないものとします。法的通知については、このベリタス製品に付随する文書、または https://www.veritas.com/about/legal/license-agreements を参照してください。

この文書に記載する製品は、使用、複製、配布、逆コンパイル/リバースエンジニアリングを制限する使用許諾の下で配布されます。この文書のいかなる部分も、Veritas Technologies LLC と、ある場合はその実施権許諾者の、事前の書面による承諾なしに、いかなる形態でいかなる手段によっても、複製されることはありません。

この文書は "現状のまま" として提供され、すべての明示的または暗示的な条件、表現、および保証(商品性、特定目的への適合性、または非侵害に関するあらゆる暗示的な保証を含む)に関する責任は、法的に無効と見なされる免責の場合を除き、免除されます。Veritas Technologies LLC は、この文書の供給、履行、または使用に関連して付随的または間接的に起こる損害に対して責任を負いません。この文書に含まれる情報は予告なしに変更することがあります。

ライセンスソフトウェアおよびマニュアルは、FAR 12.212 の規定によって商業用コンピュータソフトウェアと見なされ、ベリタス社によりオンプレミスで提供されるかホストされたサービスとして提供されるかに関わらず、FAR Section 52.227-19「Commercial Computer Software - Restricted Rights」および DFARS 227.7202「Commercial Computer Software and Commercial Computer Software Documentation」、その他の後継規制の規定により制限された権利の対象となります。使用許諾されたソフトウェアおよび文書の米国政府による修正、再生リリース、履行、表示または開示は、この契約の条件に従って行われます。

Veritas Technologies LLC
2625 Augustine Drive
Santa Clara, CA 95054
http://www.veritas.com

テクニカルサポート

テクニカルサポートはグローバルにサポートセンターを管理しています。すべてのサポートサービスは、サポート契約と現在のエンタープライズテクニカルサポートポリシーに応じて提供されます。サポート内容およびテクニカルサポートの利用方法に関する情報については、次の Web サイトにアクセスしてください。
https://www.veritas.com/support

Veritas Account 情報は、次の URL で管理できます。
https://my.veritas.com

現在のサポート契約についてご不明な点がある場合は、次に示すお住まいの地域のサポート契約管理チームに電子メールでお問い合わせください。

世界中 (日本以外)	CustomerCare@veritas.com
日本	CustomerCare_Japan@veritas.com

マニュアル

マニュアルの最新バージョンがあることを確認してください。各マニュアルには、2 ページ目に最終更新日が記載されています。最新のマニュアルは、ベリタスの Web サイトで入手できます。
https://sort.veritas.com/documents

マニュアルに関するご意見やご感想

ご意見、ご感想をお待ちしています。改善すべき点や、マニュアル上の誤記、欠落がありましたらお寄せください。お送りいただく際は、マニュアルの題名とバージョン、章のタイトル、セクションのタイトルを明記してください。フィードバックの送信先:
infoscaledocs@veritas.com

ベリタスのコミュニティサイトで、マニュアルに関する情報を確認したり、質問を投稿することもできます。
http://www.veritas.com/community/

Veritas Services and Operations Readiness Tools (SORT)

Veritas SORT (Services and Operations Readiness Tools) は、時間のかかる特定の管理タスクを自動化および単純化するための情報とツールを提供するWebサイトです。製品に応じて、SORTはインストールとアップグレードの準備、データセンターのリスクの識別、効率性の改善に役立ちます。使用している製品に対してSORTが提供しているサービスおよびツールについては、次のデータシートを参照してください。
https://sort.veritas.com/data/support/SORT_Data_Sheet.pdf

目次

第 1 部 Oracle データベース用 SFHA (Storage Foundation High Availability) 管理ソリューション 18

第 1 章 Storage Foundation for Databases の概要 19

Oracle 用 Storage Foundation High Availability（SFHA）Solutions の概要 19
Veritas File System について 20
Veritas File System のインテントログについて 21
エクステントについて 22
ファイルシステムのディスクレイアウトについて 22
Veritas Volume Manager の概要 23
DMP (Dynamic Multi-Pathing) について 24
Cluster Server について 24
Cluster Server のエージェントについて 25
Veritas InfoScale Operations Manager について 25
Veritas InfoScale 7.4.2 製品の Oracle サポート機能 26
Veritas InfoScale 製品の使用例 28

第 2 部 Veritas InfoScale 製品を使用した Oracle の配備 33

第 2 章 Storage Foundation 環境への Oracle オプションの配備 34

Storage Foundation 環境の Oracle の配備オプション 34
Storage Foundation 環境での Oracle 単一インスタンス 35
Storage Foundation 環境でオフホストを使う単一インスタンスの Oracle 36
Storage Foundation High Availability を用いた高可用性クラスタの単一インスタンス Oracle 37
SF Cluster File System HA 環境の並列クラスタの Oracle の単一インスタンス 39

Storage Foundation for Oracle RAC 環境の Oracle RAC について 41
レプリケーションされた Storage Foundation 環境での Oracle について 42
仮想化環境での Oracle と Storage Foundation の配備 43
Storage Foundation SmartMove およびシンプロビジョニングによる Oracle の配備 44

第 3 章 Storage Foundation を使用した Oracle の配備 45

Oracle データベースを配備するためのタスク 45
Oracle のストレージの計画 46
Oracle の配備のためのボリュームレイアウトの選択について 47
Oracle を配備するためのディスクグループの設定 47
Oracle を配備するためのディスクグループ設定のガイドライン 49
Oracle を配備するためのボリュームの作成 49
Oracle を配備するためのボリューム設定のガイドライン 50
Oracle を配備するための VxFS ファイルシステムの作成 50
Oracle を配備するためのファイルシステムの作成ガイドライン 51
Oracle を配備するためのファイルシステムのマウント 52
Oracle のインストールとデータベースの作成 53

第 4 章 Storage Foundation を使用したオフホスト設定での Oracle の配備 54

データベースのオフホスト設定の必要条件 54

第 5 章 High Availability を使用した Oracle の配備 56

Oracle を HA 設定に配備するためのタスク 56
データベースを高可用性にする VCS 設定 56

第 6 章 ディザスタリカバリ用 VVR (Volume Replicator) を使用した Oracle の配備 57

ディザスタリカバリ用の VVR を備えた Oracle の配備について 57
ディザスタリカバリ用の VVR を備えた Oracle を配備するタスク 58
Oracle を配備するためのプライマリノードの設定 58
Oracle を配備するためのセカンダリノードの設定 59
Oracle を配備するためのプライマリとセカンダリのノード間の VVR の設定 61

第 3 部 Storage Foundation for Databases (SFDB) ツールの設定 63

第 7 章 Storage Foundation for Databases リポジトリデータベースの設定および管理 64

SFDB (Storage Foundation for Databases)リポジトリについて 64
Storage Foundation for Databases (SFDB) ツールの必要条件 65
SFDB（Storage Foundation for Database）ツールの可用性 65
Storage Foundation for Databases (SFDB) リポジトリの設定 66
SFDB のリポジトリの場所 66
SFDB リポジトリの設定 68
Storage Foundation for Databases（SFDB）リポジトリのバックアップと復元 69
ノード追加後の SFDB（Storage Foundation for Databases）リポジトリの更新 71
ノード削除後の SFDB（Storage Foundation for Databases）リポジトリの更新 72
SFDB (Storage Foundation for Databases) リポジトリの削除 72

第 8 章 Storage Foundation for Databases (SFDB) ツールの認証の設定 74

SFDB ツール認証のための vxdbd の設定 74
SFDB ツールに対して認証を使っているクラスタへのノードの追加 75
SFDB コマンドを実行できるようにするためのユーザーの許可 76

第 4 部 Oracle データベースのパフォーマンスの向上 78

第 9 章 データベースアクセラレータについて 79

Veritas InfoScale 製品コンポーネントのデータベースアクセラレータについて 79

第 10 章 Veritas Extension for Oracle Disk Manager によるデータベースパフォーマンスの向上 82

Veritas InfoScale 製品環境における Oracle Disk Manager について 82
SFHA 環境の Oracle Disk Manager によるデータベースパフォーマンスの改善方法 84

Oracle Disk Manager と Oracle Managed Files の連携 86
SFHA 環境の Veritas Extension for Oracle Disk Manager の設定 88
SFHA 環境の Veritas Extension for Oracle Disk Manager の設定 89
SFHA 環境で既存のデータベースストレージを Oracle Disk Manager 用として準備する方法 91
SFHA 環境に Oracle Disk Manager を設定していることを確認 92
SFHA 環境の Oracle Disk Manager 機能の無効化 94

第 11 章 Veritas Cached Oracle Disk Manager によるデータベースパフォーマンスの向上 96

SFHA 環境の Cached ODM について 96
SFHA 環境の Cached ODM の使用に関する注意事項 96
SFHA 環境の Cached ODM でサポートされている設定 97
SFHA 環境の Cached ODM Advisor について 97
SFHA 環境の Cached ODM の設定 98
SFHA 環境のファイルシステムに対する Cached ODM の有効化 98
SFHA 環境の個々のファイルの Cached ODM 設定の調整 99
SFHA 環境の cachemap を使った Cached ODM 設定のチューニング 100
SFHA 環境のマウント全体を通したキャッシュ設定の永続化 100
SFHA 環境の Cached ODM Advisor による Cached ODM 設定の管理 101
SFHA 環境の Cached ODM Advisor を使用した候補データファイルのレポートの生成 104
SFHA 環境の Cached ODM Advisor を使用したデータファイルの Cached ODM の有効化/無効化 109
SFHA 環境の Cached ODM Advisor を使用したデータベースファイルの Cached ODM 状態の表示 111
SFHA 環境の Cached ODM Advisor を使用した Cached ODM 統計情報の表示 112
SFHA 環境の Cached ODM Advisor を使用した ODM I/O 統計情報の表示 113

第 12 章 Quick I/O によるデータベースパフォーマンスの向上 115

Quick I/O について 115
Quick I/O によるデータベースパフォーマンス向上のしくみ 116
qiomkfile による Quick I/O ファイルとしての Oracle データベースファイルの作成 118
setext コマンドによる Quick I/O ファイル領域の事前割り当て 121

通常の VxFS ファイルへの Quick I/O ファイルとしてのアクセス 122
Oracle ファイルの Quick I/O ファイルへの変換 124
スパースファイルについて 129
Oracle データベースにおける一時表領域と Quick I/O の扱い 130
Quick I/O の状態とファイル属性の表示 132
Quick I/O ファイルの拡張 133
Quick I/O ファイルでの Oracle AUTOEXTEND の使用 135
データベースのリストア後の Quick I/O ファイルの再作成 137
Quick I/O の無効化 139
Solaris ローカルゾーンへの Quick I/O ファイルの作成 140

第 13 章 Cached Quick I/O によるデータベースパフォーマンスの向上 142

Cached Quick I/O について 142
Cached Quick I/O 設定のタスク 143
ファイルシステム上での Cached Quick I/O の有効化 143
qio_cache_enable フラグの有効化および無効化 144
再起動およびマウントにわたる Cached Quick I/O 設定の永続化 145
vxtunefs によるチューニング情報の取得 146
Cached Quick I/O 対象ファイルの判断 148
I/O 統計情報の収集 148
Oracle の I/O 統計情報について 149
I/O 統計値上での先読みの影響 151
その他の分析ツール 151
個々のファイルに対する Cached Quick I/O の有効化と無効化 151
個別のファイルに対するキャッシュアドバイザリの設定 152
個々のファイルの Cached Quick I/O 設定の永続化 153
qioadmin による個々のファイルの Cached Quick I/O 設定の判定 153

第 5 部 PITC の使用 155

第 14 章 PITC 方法の理解 156

PITC（Point-In-Time Copy）の概要 156
PITC を使う状況 157
Storage Foundation PITC テクノロジについて 158
SFDB ツールによってサポートされている PITC ソリューション 159
Storage Foundation for Databases (SFDB) ツールのサポート対象スナップショットモードについて 160
ボリュームレベルのスナップショット 162

ボリュームスナップショットの永続 FastResync 162
ボリュームスナップショットのデータ整合性 163
サードミラーブレークオフスナップショット 163
領域最適化インスタントスナップショット 164
ボリュームレベルのスナップショット（FlashSnap）の逆再同期について 165
逆再同期操作のタイプ 166
逆再同期の制限事項 167
逆再同期操作を実行するための前提条件 168
5.X から逆再同期の実装に導入された変更 168
逆再同期操作のパラメータの設定 168
Storage Checkpoint 170
Storage Checkpoint とスナップショットの違い 170
Storage Checkpoint の動作 171
Storage Checkpoint を使用したデータベースのロールバックについて 175
Checkpoint と Storage Rollback のプロセス 176
Storage Checkpoint の種類 177
Storage Checkpoint の領域管理に関する注意事項 178
FileSnap について 179
FileSnap のプロパティ 179
FileSnap に対する同時 I/O 180
コピーオンライトと FileSnap 180
FileSnap からの読み取り 181
ブロックマップの断片化と FileSnap 181
バックアップと FileSnap 181

第 15 章 Oracle PITC に関する注意事項 183

データベースレイアウトの注意事項 183
サポート対象の Oracle 設定 184
マルチテナントデータベースのサポートに関する考慮事項 185
クローンデータベースの動作 186

第 16 章 サードミラーブレークオフスナップショットの管理 187

クローン作成のための Database FlashSnap 187
Database FlashSnap の利点 188
Database FlashSnap のホストとストレージの準備 188
ホストの設定 188
データベースによって使われるボリュームまたはボリュームセットのスナップショットミラーの作成 190
Database FlashSnap を使用したデータベースのクローンの作成 194
ミラーボリュームのプライマリボリュームとの再同期 200

スナップショットボリュームを逆再同期するには 202
逆再同期パラメータの出力例 202
クローンデータベースの手動リカバリ 204
セカンダリホストでのデータベースのクローン作成 205

第 17 章 領域最適化スナップショットの管理 207

インスタント領域最適化スナップショットの作成の計画 207
インスタントスナップショットを作成するための従来のボリュームの準備 208
共有キャッシュオブジェクトの作成 209
領域最適化スナップショットを使った Oracle データベースのクローンの作成 210
HA 環境の領域最適化スナップショット 214
FlashSnap スナップショットを使った複数のクローン作成 215
クローンデータベースの手動リカバリ 221

第 18 章 Storage Checkpoint の管理 222

Storage Checkpoint について 222
リカバリのための Database Storage Checkpoint 223
Database Storage Checkpoint の利点と制限事項 224
Database Storage Checkpoint の作成 224
Database Storage Checkpoint の削除 225
Database Storage Checkpoint のマウント 225
Database Storage Checkpoint のマウント解除 226
Database Storage Checkpoint を使ったデータベースのクローンの作成 227
Database Storage Checkpoint からのリストア 227
オフラインモード Storage Checkpoint のデータの収集 229

第 19 章 FileSnap スナップショットの管理 230

FileSnap を使う準備 230
FileSnap を使った Oracle データベースのクローンの作成 231
FileSnap を使用して作成したデータベースのクローンの破棄 233
FileSnap スナップショットの破棄 234
FileSnap スナップショットからのデータベースファイルの復元 235
FileSnap を使って作成したスナップショットとクローンの一覧表示 236

第 20 章 SFHA 環境での Netbackup によるバックアップとリストア 237

Veritas NetBackup について 237
Oracle データベースファイルに対する NetBackup の BLI バックアップの動作 238
Veritas NetBackup を使った Oracle のバックアップとリストアについて 238
Veritas NetBackup を使った Oracle Disk Manager ファイルのバックアップとリストアについて 239
NetBackup を使った Oracle 用 Quick I/O ファイルのバックアップとリストア 239
SFHA Solutions 製品環境での NetBackup の使用 241
NetBackup マスターサーバーのクラスタ化 241
NetBackup を使用した VxVM ボリュームのバックアップとリカバリ 242
NetBackup を使った VxVM ボリュームのリカバリ 243
NetBackup を使用した Oracle データベースのバックアップ 243
NetBackup を使用した Oracle データベースの回復 244

第 6 部 Oracle に対するストレージコストの最適化 246

第 21 章 SmartTier によるストレージの階層化を理解する 247

SmartTier について 247
VxFS MVS ファイルシステムについて 249
VxVM ボリュームセットについて 250
ボリュームタグについて 250
SmartTier ファイルの管理 250
SmartTier のサブファイルオブジェクトの管理 251
SmartTier for Oracle の動作方法 251
パーティションによる Oracle のデータベースオブジェクトの格納方法の変更 251
データベースエクステントのファイル位置 255
Oracle の統計情報 255
Oracle データベースオブジェクトのストレージの最適化 256
Automatic Workload Repository の統計情報について 257
高可用性（HA）環境での SmartTier 258

第 22 章 SmartTier の設定と管理 259

SmartTier for Oracle の設定 259
SmartTier for Oracle コマンドの必要条件 260
データベースパラメータの定義 260
ストレージクラスの設定 263
Veritas File System（VxFS）から VxFS MVS ファイルシステムへの変換 265
ボリュームのストレージクラスへの分類 267
ストレージクラスの空き領域の表示 267
ストレージクラスへの新しいボリュームの追加 269
ストレージクラスからのボリュームの削除 270
SmartTier for Oracle を使ったデータベースストレージの最適化 271
SmartTier for Oracle を使ったファイルの手動再配置 271
SmartTier for Oracle を使った表領域の手動再配置 272
SmartTier for Oracle を使ったテーブルパーティションの手動再配置 273
SmartTier for Oracle を使ったストレージ階層化を自動化するためのポリシー設定 274
SmartTier for Oracle を使用したレポートの実行 275
SmartTier for Oracle を使ったデータベース環境でのエクステント分散 277
データベースパフォーマンスを向上させるためのエクステント分散ファイルシステム 278
SmartTier for Oracle を使ったエクステント分散ファイルシステムの作成 279
SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトのレポートの実行 281
SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトの配置の最適化 283

第 23 章 Oracle での SmartTier のユースケース 286

Oracle での SmartTier の使用例 286
SmartTier for Oracle のファイルベースの使用例 286
パーティション分割されたデータと表領域の移行 287
アーカイブログと Flashback ログの再配置のスケジュール 289
SmartTier for Oracle サブファイルの使用例 291
AWR の統計情報を使った、再配置の可能性があるサブファイルオブジェクトの特定 292
上位のパフォーマンス階層へのホットデータベースセグメントの再配置 294
低コスト階層またはセカンダリ階層への古いテーブルデータの再配置 295

第 24 章 ストレージコストを最適化するためのファイルとデータベースの圧縮 297

圧縮ファイルとデータベース 297
サポートされるデータベースバージョンと環境 298
アーカイブログの圧縮 298
読み取り専用表領域の圧縮 299
アクセス頻度が低いテーブルパーティションの圧縮 300
アクセス頻度が低いデータファイルの圧縮 301
Oracle データベースのファイルの圧縮の推奨設定 302

第 25 章 圧縮アドバイザツールの使用 303

圧縮アドバイザツールについて 303
圧縮アドバイザを使った Oracle アーカイブログの圧縮 304
圧縮アドバイザを使った圧縮の候補データファイルの表示 305
圧縮アドバイザを使った Oracle データファイルの圧縮 306
圧縮アドバイザを使った圧縮データファイルの表示 307
圧縮アドバイザを使った Oracle データファイルの圧縮解除 308
圧縮アドバイザのコマンドリファレンス 308

第 7 部 Oracle ディザスタリカバリの管理 311

第 26 章 レプリケート環境での Storage Foundation for Databases (SFDB) ツールの使用 312

SF Databases ツールと VVR（Volume Replicator）の配備について 312
VVR（Volume Replicator）環境での Oracle データベースに対するデータベースアクセラレータの使用 313
Oracle Data Guard での Storage Foundation for Databases（SFDB）ツールの配備について 314

第 27 章 レプリケート環境でのボリュームスナップショットの使用 316

ボリュームスナップショットを VVR と併用する場合の考慮事項 316
VVR セットアップでの領域最適化スナップショットの作成 317
VVR セットアップでのサードミラーブレークオフスナップショットの作成 320
レプリケーションロールの転送後に vxsfadm を使用するための注意事項 324

第 8 部 Storage Foundation for Databases 管理リファレンス 325

第 28 章 Storage Foundation for Databases コマンドリファレンス 326

vxsfadm コマンドリファレンス 326
FlashSnap リファレンス 329
FlashSnap の設定パラメータ 329
FlashSnap がサポートされる操作 332
領域最適化スナップショットリファレンス 333
領域最適化スナップショットの設定パラメータ 333
領域最適化スナップショットがサポートされる操作 335
FileSnap リファレンス 336
FileSnap 設定パラメータ 336
FileSnap がサポートされる操作 337
Database Storage Checkpoints のリファレンス 338
Database Storage Checkpoint の設定パラメータ 338
Database Storage Checkpoint のサポートされる操作 340

第 29 章 Storage Foundation for Databases のチューニング 343

その他のマニュアル 343
VxVM のチューニングについて 344
ボリュームの I/O 統計情報の取得について 344
VxFS のチューニングについて 345
空き領域の監視の動作 346
VxFS I/O パラメータのチューニングの動作 347
VxFS I/O チューニングパラメータについて 347
Quick I/O インターフェースを使用したファイル I/O の統計情報の取得について 352
I/O 統計データについて 353
Veritas extension for Oracle Disk Manager を使用したファイル I/O 統計の取得 355
I/O 統計について 356
Oracle データベースのチューニングについて 357
順次テーブルスキャン 357
順次テーブルスキャン 358
非同期 I/O 358
バッファキャッシュのチューニング 359
バッファキャッシュのチューニング 359

順次スキャン中の Oracle のブロック読み取りの設定 359
スレーブパラメータの設定 361
メモリ割り当ての設定 361
AIX の仮想メモリマネージャのチューニングについて 362
Oracle 用の Solaris のチューニングについて 365
maxuprc 365
shmmax 366
shmmin 366
shmmni 366
shmseg 366
semmap 366
semmni 366
semmns 366
semmnu 367
semmsl 367

第 30 章 SFDB ツールのトラブルシューティング 368

Storage Foundation for Databases（SFDB）ツールのトラブルシューティングについて 368
SFDB ツールのテクニカルサポート分析用スクリプトの実行 369
Storage Foundation for Databases（SFDB）ツールのログファイル 369
vxdbd デーモンについて 369
vxdbd の起動と停止 369
vxdbd デーモンのリスニングポートの設定 370
vxdbd リソース使用量の制限 370
vxdbd の暗号化暗号文の設定 371
vxdbd のトラブルシューティング 371
SFDB ツールのトラブルシューティングのためのリソース 372
SFDB ログ 372
SFDB エラーメッセージ 373
SFDB リポジトリとリポジトリファイル 373
SmartTier for Oracle のトラブルシューティング 374
SFDB（Storage Foundation for Databases）ツールの 5.0x から 7.4.2 へのアップグレード（2184482） 375
逆再同期のトラブルシューティング 376

第 31 章 Oracle データベースの手動によるリカバリ 378

クローンデータベースの手動による回復 378
復元操作の後の手動による回復 379

第 32 章 6.0 より前のリリースの Storage Foundation for Databases のコマンドリファレンス 381

SFDB コマンドの後方互換について 381
サポート対象外となった Storage Foundation for Databases（SFDB）ツールの機能 382
Database FlashSnap のストレージの準備 382
ボリュームのスナップショットミラーを作成する手順例 382
Veritas Volume Manager 7.4.2 を使うための既存ボリュームのアップグレード 386
データベースの構造変更後の更新 393
データベーススナップショットの作成について 394
オンラインデータベーススナップショット 394
スナップショット作成前の作業 395
スナップショットの作成 396
スナップショット作成後の作業 398
FlashSnap コマンド 403
スナッププランの作成（dbed_vmchecksnap） 404
スナッププランの検証（dbed_vmchecksnap） 414
スナッププランの表示、コピー、削除（dbed_vmchecksnap） 416
スナップショットの作成（dbed_vmsnap） 419
スナップショットボリュームを使ったデータベースのバックアップ（dbed_vmclonedb） 422
クローンデータベースの作成（dbed_vmclonedb） 426
スナップショットとデータベースの再同期 433
スナップショットボリュームの削除 435
Oracle リカバリのガイドライン 436
Storage Rollback 前の全制御ファイルのバックアップ 436
制御ファイルがロールバックされないことの確認 436
すべてのアーカイブ REDO ログが利用可能であることの確認 437
メディアのリカバリ手順 437
Database Storage Checkpoint のコマンド 439
dbed_update を使ったリポジトリの作成または更新 440
dbed_ckptcreate を使った Storage Checkpoint の作成 441
dbed_ckptdisplay を使った Storage Checkpoint の表示 443
dbed_ckptmount を使った Storage Checkpoint のマウント 444
dbed_ckptumount を使った Storage Checkpoint のマウント解除 446
dbed_ckptrollback を使った Storage Rollback の実行 446
dbed_ckptremove を使った Storage Checkpoint の削除 448
dbed_clonedb を使った Oracle インスタンスのクローン作成 448

第 9 部 参照先 452

付録 A VCS Oracle エージェント 453

Oracle 用の VCS エージェント 453
Oracle エージェント関数 453
Oracle エージェントのリソースタイプ定義 457
Netlsnr エージェント関数 462
Netlsnr エージェントのリソースタイプ定義 463

付録 B クラスタ化された配備のサンプル設定ファイル 466

サンプル設定ファイルについて 466
Storage Foundation for High Availability のサンプル設定ファイル 467
Storage Foundation for Clustered File System HA のサンプル設定ファイル 469
Storage Foundation for Oracle RAC のサンプル設定ファイル 475

付録 C Database FlashSnap の状態情報 481

Database FlashSnap の状態情報 481
CLI からの Database FlashSnap スナップショットの状態情報 481
CLI からのスナップショットの状態情報 482
CLI からのスナップショットデータベースの状態情報 484

付録 D サードパーティソフトウェアによるファイルのバックアップ 485

サードパーティソフトウェアを使ったファイルのバックアップについて 485
サードパーティソフトウェアによるファイルのバックアップ 485
Oracle RMAN を使った Oracle Disk Manager ファイルのバックアップとリストア 485

1

Oracle データベース用 SFHA (Storage Foundation High Availability) 管理ソリューション

- 第1章 Storage Foundation for Databases の概要

1

Storage Foundation for Databases の概要

この章では以下の項目について説明しています。

- Oracle 用 Storage Foundation High Availability（SFHA）Solutions の概要
- Veritas File System について
- Veritas Volume Manager の概要
- DMP (Dynamic Multi-Pathing) について
- Cluster Server について
- Cluster Server のエージェントについて
- Veritas InfoScale Operations Manager について
- Veritas InfoScale 7.4.2 製品の Oracle サポート機能
- Veritas InfoScale 製品の使用例

Oracle 用 Storage Foundation High Availability（SFHA）Solutions の概要

このマニュアルでは、Oracle データベース環境における Storage Foundation High Availability（SFHA）Solutions 製品での SFDB ツールの配備と使用について主要な例を示します。これは、SFHA Solutions 製品のマニュアルと共に参照するための補足マニュアルです。

Storage Foundation for Databases ツールは、Oracle データベースの管理を強化するオプションを提供します。SFDB ツールには root 権限のないデータベース管理者でも実行できる使いやすい強化されたコマンドがあり、Oracle データベース環境でストレージ

を最適化できます。このマニュアルでは、SFHA Solutions のエンタープライズ製品での SFDB ツールの配備と使用について説明します。

メモ: このマニュアル内で RHEL (Red Hat Enterprise Linux) オペレーティングシステムについて使用するコマンドは、サポート対象の RHEL 互換配布にも適用されます。

Veritas File System について

ファイルシステムとは、コンピュータファイルとそれに含まれるデータを保存、整理する単純な手法で、検索とアクセスを容易にします。形式的に言えば、ファイルシステムとは、データのストレージ、階層編成、操作、ナビゲーション、アクセス、取り込みなどが実装された抽象データ型（メタデータなど）の一式です。

Veritas File System（VxFS）は、商用では初めてのジャーナルファイルシステムです。ジャーナルを使うと、メタデータの変更はまずログ（またはジャーナル）に書き込まれ、次にディスクに書き込まれます。変更を複数の場所に書き込む必要がなく、メタデータは非同期で書き込まれるため、スループットが大幅に速くなります。

また、VxFS は、エクステントを管理単位としたインテントログファイルシステムです。VxFS は、優れたパフォーマンスと高い可用性、そして大容量のデータ処理能力を必要とする操作環境にとって最適なファイルシステムです。

作成できるファイルシステムの最大サイズは、ブロックサイズによって異なります。

ブロックサイズ	現在サポートされているファイルシステムの最大サイズ
1024 バイト	68,719,472,624 セクタ（≈32 TB）
2048 バイト	137,438,945,248 セクタ（≈64 TB）
4096 バイト	274,877,890,496 セクタ（≈128 TB）
8192 バイト	549,755,780,992 セクタ（≈256 TB）

VxFS の主要コンポーネントは次のとおりです。

ファイルシステムのログ	「Veritas File System のインテントログについて」
エクステント	「エクステントについて」
ファイルシステムのディスクレイアウト	「ファイルシステムのディスクレイアウトについて」

Veritas File System のインテントログについて

通常のファイルシステムはシステム障害からリカバリする際、そのための唯一の手段である fsck ユーティリティによるファイルシステム構造全体の検証に依存しています。ディスク構成の規模が大きい場合には、このユーティリティによる構造全体の調査、ファイルシステムの整合性の検証および不整合部分の修正の処理には、必然的に相当の時間が必要になります。 VxFS は、VxFS インテントログと VxFS インテントログのサイズ変更機能で高速リカバリを実現します。

VxFS では、VxFS インテントログでファイルシステムの動作を追跡することによって、システム障害からのリカバリ時間が短縮されます。この機能は、ファイルシステム構造に加えられた変更のうち保留中になっているものを、循環式のインテントログに記録します。システム障害の場合を除き、ユーザーやシステム管理者がインテントログによるリカバリ機能に気付くことはありません。デフォルト設定時の VxFS ファイルシステムでは、ファイルのトランザクションをディスクにコミットする前にログに記録し、ファイルシステムが予想に反して停止した場合にファイルシステムのリカバリに要する時間を短縮させます。

システム障害からのリカバリ中、VxFS の fsck ユーティリティはインテントログの再生（インテントログのスキャン）を実行し、システム障害時に実行されていたファイルシステム操作を取り消すか、または完了させます。そのため、ファイルシステムの構造全体を検査しなくてもファイルシステムをマウントできる状態になります。ディスクにハードウェア障害があると、インテントログを再生しても損傷を受けたファイルシステム構造のリカバリを完了できない場合があります。そのような場合は、VxFS の fsck ユーティリティを使って、ファイルシステムの構造全体の検査を完了させる必要があります。

mount コマンドは、ファイルシステムでダーティログを検出すると、VxFS の fsck コマンドを自動的に実行してインテントログの再生を実行します。この機能は、Veritas Volume Manager（VxVM）ボリュームにマウントされたファイルシステムでのみサポートされ、クラスタファイルシステムでサポートされます。

fsck_vxfs（1M）マニュアルページとmount_vxfs（1M）マニュアルページを参照してください。

VxFS インテントログは、ファイルシステムを最初に作成する際に割り当てられます。インテントログのサイズは、ファイルシステムのサイズによって異なり、ファイルシステムが大きくなるほど、インテントログのサイズも大きくなります。 fsadm コマンドを使ってインテントログのサイズを後から調整できます。

fsadm_vxfs (1M) のマニュアルページを参照してください。

ディスクレイアウトバージョン 7 以降のインテントログの最大デフォルトサイズは 256 MB です。

メモ: インテントログを不適切なサイズに設定すると、システムの処理効率に悪影響を及ぼす可能性があります。

エクステントについて

エクステントは、コンピュータファイルシステム内の連続したストレージ領域で、ファイル用に予約された領域です。ファイルに対する書き込みを開始すると、エクステント全体が割り当てられます。ファイルに再び書き込みを行うと、前回の書き込み場所に続けてデータが書き込まれます。これにより、ファイルの断片化を少なくするかまたは回避できます。エクステントは「アドレスと長さの組み合わせ」で表現され、これによって開始ブロックのアドレスと、(ファイルシステムまたは論理ブロックの)エクステントの長さが決まります。Veritas File System (VxFS) は、エクステントベースのファイルシステムのため、エクステント(複数ブロックで構成可能)を使ってアドレス指定します。単一ブロックセグメントにアドレス指定するのではありません。したがって、エクステントを使うとファイルシステムのスループットを向上できます。

エクステントを使った場合、格納領域に連続するブロックが割り当てられると、複数のブロック単位でディスク I/O が可能になります。順次 I/O の場合、複数ブロック単位での処理の方が 1 ブロック単位のものよりもかなり高速になり、ほぼすべてのディスクドライブで複数ブロックの I/O 操作を実行できます。

エクステント単位の割り当ての場合、i ノード構造体からアドレスブロック情報を解釈する方法が、ブロック単位の割り当ての場合と比較して多少異なります。VxFS の i ノードは、10 個のエクステントを直接参照し、各エクステントは開始ブロックアドレスとブロック単位での長さの組み合わせで表現されています。

ディスク領域としては 512 バイトのセクタが割り当てられ、このセクタから論理ブロックが構成されます。VxFS では、1024、2048、4096、8192 バイトの論理ブロックサイズがサポートされています。デフォルトのブロックサイズは、2 TB 未満のファイルシステムに対しては 1 KB、2 TB 以上のファイルシステムに対しては 8 KB です。

ファイルシステムのディスクレイアウトについて

ディスクレイアウトは、ファイルシステムの情報がディスクに保存される方法です。VxFS (Veritas File System) では、新機能や特定の UNIX 環境を提供するため、さまざまなディスクレイアウトバージョンがサポートされています。

ディスクレイアウトバージョンをアップグレードするには、次のいずれかのコマンドを使用できます。

`vxupgrade`

ファイルシステムがオンラインのときに、既存の VxFS ファイルシステムを、サポート対象のディスクレイアウトバージョンにアップグレードします。

`vxupgrade`(1M) マニュアルページを参照してください。

`vxfsconvert`

ファイルシステムがマウントされていないときに、サポート対象外のディスクレイアウトバージョンを、サポート対象のバージョンにアップグレードします。

`vxfsconvert` コマンドは、ファイルシステムがマウントされていないときにネイティブファイルシステム (ext2、ext3、および ext4) を VxFS に変換する場合にも使用できます。

`vxfsconvert`(1M) マニュアルページを参照してください。

表 1-1は、サポート対象のディスクレイアウトバージョンを一覧表示します。

表 1-1　サポート対象のディスクレイアウトバージョン

バージョン	サポート対象機能
バージョン 12	バージョン 12 は CFS で 128 ノードクラスタをサポートします。
バージョン 13	■ WORM のサポートの追加 ■ 拡張ファイル属性使用時のクローン作成パフォーマンスの向上
バージョン 14	SmartIO FEL ベースのキャッシュをサポートします。
Version 15	■ SELinux 属性のストレージと取得のパフォーマンスの向上 ■ WORM ファイルの安全なクロックサポート
Version 16	WORM ファイルの監査ログのサポート

現在作成およびマウントできるのは、バージョン 12、13、14、15、16 のみです。バージョン 6、7、8、9、10、11 はマウントできますが、サポートされているバージョンにアップグレードする場合のみです。

Veritas Volume Manager の概要

Veritas 社の Veritas™ Volume Manager (VxVM) は、物理ディスクや論理ユニット番号 (LUN) をボリュームと呼ばれる論理デバイスとして管理できるようにする、ストレージ管理サブシステムです。VxVM ボリュームは、ファイルシステム、データベース、その他の管理対象データオブジェクトの設定ができる物理デバイスとして、アプリケーションとオペレーティングシステムに表示されます。

VxVM は、コンピュータ環境とストレージエリアネットワーク (SAN: Storage Area Network) 環境で、使いやすいオンラインディスクストレージ管理を実現します。RAID (Redundant Array of Independent Disks) をサポートすることにより、VxVM はディスクやハードウェアの障害からシステムを保護し、I/O スループットを高めるように設定できます。さらに VxVM には、耐障害性およびディスク障害またはストレージアレイ障害からの迅速なリカバリを拡張する機能があります。

VxVM では、論理ボリューム管理レイヤーによって、ハードウェアディスクデバイスや LUN からの制約を克服できます。このため、ボリュームを複数のディスクと LUN に分散できます。

VxVM は、パフォーマンスを高め、データの可用性と整合性を確保するツールを提供しています。また、VxVM を使って、システムがアクティブなときにストレージを動的に設定できます。

DMP (Dynamic Multi-Pathing) について

Dynamic Multi-Pathing (DMP) は、システム上で設定されているオペレーティングシステムのネイティブデバイスに対するマルチパス機能を提供します。DMP は DMP メタデバイス (既知の DMP ノード) を作成して、同じ物理 LUN へのデバイスパスをすべて示します。

DMP メタデバイスは OS ネイティブの論理ボリュームマネージャ（LVM）をサポートします。DMP メタデバイス上に LVM ボリュームとボリュームグループを作成できます。

DMP はページング装置として使う LVM ボリュームデバイスをサポートします。

DMP メタデバイスは ZFS をサポートします。DMP メタデバイス上に ZFS プールを作成できます。DMP は、ルートと非ルートの ZFS プールの両方をサポートします。Solaris 11 update 1 以前のバージョンでは、DMP は非ルートの ZFS ファイルシステムのみをサポートします。

Veritas Volume Manager (VxVM) ボリュームとディスクグループは LVM ボリュームとボリュームグループと共存させることができます。ただし、各デバイスでサポートできる形式は 1 つだけです。ディスクに VxVM ラベルが付いている場合、そのディスクは LVM で利用できません。同様に、ディスクが LVM によって使用中の場合、そのディスクは VxVM で利用できません。

Veritas Volume Manager (VxVM) ボリュームとディスクグループは ZFS プールと共存できますが、各デバイスは 1 つのタイプしかサポートできません。ディスクに VxVM ラベルが付いている場合、そのディスクは ZFS で利用できません。同様に、ディスクが ZFS によって使用中の場合、そのディスクは VxVM で利用できません。

Cluster Server について

VCS（Cluster Server）は次の利点を提供するクラスタソリューションです。

- ダウンタイムを最小化する。
- サーバーの統合とフェールオーバーを促進する。
- 異機種混合環境の広い応用範囲を効果的に管理する。

製品をインストールする前に『Veritas InfoScale 7.4.2 リリースノート』を参照してください。

製品を設定するには、『Cluster Server 汎用アプリケーションエージェント設定ガイド』の指示に従います。

Cluster Server のエージェントについて

Veritas InfoScale エージェントは特定のリソースとアプリケーションに高可用性を提供します。各エージェントは特定のタイプのリソースを管理します。たとえば、Oracle のエージェントは Oracle のデータベースを管理します。通常、エージェントはリソースの開始、停止、監視を行い、状態の変化を報告します。

VCS エージェントをインストールする前に、そのエージェントの設定ガイドを参照してください。

このリリースで提供されるエージェントに加えて、他のエージェントは別売のオプション製品（Cluster Server Agent Pack）によって利用可能です。エージェントパックは現在発売されているエージェントを含んでおり、現在開発中の新しいエージェントを追加するために年 4 回再リリースされます。

次の詳細については Veritas InfoScale の販売担当者にお問い合わせください:

- エージェントパックに含まれるエージェント
- 開発中のエージェント
- Veritas InfoScale のコンサルティングサービスから利用可能なエージェント

最新のエージェントは、Veritas Services and Operations Readiness Tools の Web サイトからダウンロードできます。

http://sort.veritas.com/agents

Veritas InfoScale Operations Manager について

Veritas InfoScale Operations Manager には、Veritas InfoScale 製品の集中型管理コンソールが用意されています。Veritas InfoScale Operations Manager を使って、ストレージリソースを監視、視覚化、管理したり、レポートを生成したりすることができます。

Veritas InfoScale Operations Manager を使って Storage Foundation と Cluster Server の環境を管理することをお勧めします。

Veritas InfoScale Operations Manager は https://sort.veritas.com/ からダウンロードできます。

インストール、アップグレード、設定の手順について詳しくは、Veritas InfoScale Operations Manager のマニュアルを参照してください。

Veritas Enterprise Administrator（VEA）のコンソールは Veritas InfoScale 製品に含まれなくなりました。VEA の使用を続けたい場合は、
https://www.veritas.com/product/storage-management/infoscale-operations-manager

からソフトウェアバージョンをダウンロードできます。Storage Foundation Management Server は非推奨です。

Veritas InfoScale 7.4.2 製品の Oracle サポート機能

Veritas InfoScale Solutions と Oracle の使用例は、SFHA（Veritas InfoScale Storage Foundation and High Availability）Solutions 製品の共有管理機能に基づいています。クラスタ化機能は、VCS（Cluster Server）と SFHA Solutions 製品から別々に使用できます。

表 1-2 に SFHA Solutions 製品をまたがってサポートされる機能をリストします。表 1-3 に VCS で利用可能な高可用性機能とディザスタリカバリ機能をリストします。

表 1-2 Veritas InfoScale 製品のストレージ管理機能

ストレージ管理機能	Veritas InfoScale Foundation	Veritas InfoScale Storage	Veritas InfoScale Availability	Veritas InfoScale Enterprise
Veritas Extension for Oracle Disk Manager	いいえ	はい **メモ:** 単一ノードからのみこの機能にアクセスできるクラスタ排他モードでサポートされる。	いいえ	はい
Veritas Extension for Cached Oracle Disk Manager **メモ:** Oracle RAC ではサポートされていません。	いいえ	はい	いいえ	はい
Quick I/O **メモ:** Linux でサポートされていない	いいえ	はい	いいえ	はい
Cached Quick I/O **メモ:** Linux でサポートされていない	いいえ	はい	いいえ	はい
圧縮	いいえ	はい	いいえ	はい
重複排除	いいえ	はい	いいえ	はい
Flexible Storage Sharing	いいえ	はい	いいえ	はい

ストレージ管理機能	Veritas InfoScale Foundation	Veritas InfoScale Storage	Veritas InfoScale Availability	Veritas InfoScale Enterprise
SmartIO **メモ:** SFRAC はライトバックキャッシュをサポートしていません。	いいえ	はい	いいえ	はい
SmartMove	いいえ	はい	いいえ	はい
SmartTier for Oracle	いいえ	はい	いいえ	はい
シン再生	いいえ	はい	いいえ	はい
Portable Data Containers	いいえ	はい	いいえ	はい
Database FlashSnap	いいえ	はい	いいえ	はい
Database Storage Checkpoint	いいえ	はい	いいえ	はい
FileSnap	いいえ	はい	いいえ	はい
ボリュームレプリケーション	いいえ	はい	いいえ	はい
ファイルレプリケーション **メモ:** Linux でのみサポートされます。	いいえ	はい	いいえ	はい
仮想ストレージの高度なサポート	はい	はい	はい	はい
高可用性（HA）を確保するクラスタ化機能	いいえ	いいえ	はい	いいえ
ディザスタリカバリ機能（HA/DR）	いいえ	いいえ	はい	いいえ
Dynamic Multi-pathing	はい	はい	はい	はい

表 1-3 Veritas InfoScale Solutions 製品の可用性管理機能

可用性管理機能	VCS HA/DR
高可用性（HA）を確保するクラスタ化	はい
データベースとアプリケーション/ISV エージェント	はい
高度なフェールオーバーのロジック	はい

可用性管理機能	VCS HA/DR
I/O フェンシングのデータ整合性の保護	はい
高度な仮想マシンサポート	はい
Virtual Business Services	はい
レプリケーションエージェント	はい
RDC（Replicated Data Cluster）	はい
キャンパス（ストレッチ）クラスタ	はい
グローバルクラスタ化（GCO）	はい
ファイアドリル	はい

注意:

- Y=機能はライセンスに含まれています。
- N=機能はライセンスでサポートされません。

注意:

- SmartTier for Oracle は DST（Dynamic Storage Tiering）を拡張し、名前を変更したものです。
- 表 1-2 と 表 1-3 に示した機能は、例外として示されているものを除き、すべて AIX、Linux、Solaris でサポートされています。サポート対象のオペレーティングシステムについて詳しくは、個別の製品マニュアルを参照してください。

Veritas InfoScale 製品の使用例

SFHA（Veritas Storage Foundation and High Availability）Solutions 製品のコンポーネントや機能は別々に使うことができますが、連携して使うとパフォーマンスと耐性が向上し、ストレージやアプリケーションの管理が容易になります。このマニュアルでは、SFHA Solutions 製品の管理機能の主な使用例について説明します。

メモ: このマニュアル内で RHEL (Red Hat Enterprise Linux) オペレーティングシステムについて使用するコマンドは、サポート対象の RHEL 互換配布にも適用されます。

表 1-4 SFHA Solutions 製品の主な使用例

使用例	Veritas InfoScale の機能
SFHA Solutions データベースアクセラレータを使用して Oracle データベースのパフォーマンスを向上させることにより、管理機能とファイルシステムの利便性を維持しながら、データベースにおいて RAW ディスクの高速化を実現します。 p.79 の「Veritas InfoScale 製品コンポーネントのデータベースアクセラレータについて」を参照してください。	Veritas Extension for Oracle Disk Manager p.82 の「Veritas InfoScale 製品環境における Oracle Disk Manager について」を参照してください。 Veritas Extension for Cached Oracle Disk Manager p.96 の「SFHA 環境の Cached ODM について」を参照してください。 **メモ:** Oracle RAC では、Cached ODM はサポートされていません。 Quick I/O p.115 の「Quick I/O について」を参照してください。 Cached Quick I/O p.142 の「Cached Quick I/O について」を参照してください。 **メモ:** Quick I/O と Cached Quick I/O は Linux ではサポートされていません。
SFHA Solutions Flashsnap、Storage Checkpoint、NetBackup PITC 方法を使用してデータのバックアップおよびリカバリを行うことにより、データを保護します。 p.156 の「PITC（Point-In-Time Copy）の概要」を参照してください。	FlashSnap Storage Checkpoint SFHA Solutions による NetBackup
SFHA Solutions ボリュームスナップショットを使用してデータをオフホストで処理することにより、実働ホストのパフォーマンス低下を防止します。	FlashSnap
SFHA Solutions の PITC 方法を使用して、テスト、決定モデリング、開発を目的として実働データベースのコピーを最適化します。	FlashSnap

使用例	Veritas InfoScale の機能
PITC コピーにおいてファイルシステムやボリュームよりも詳細なレベルの処理が必要となる場合に、SFHA Solutions の領域最適化 FileSnap を使用して、ファイルレベルの PITC スナップショットを作成します。FileSnap は仮想マシンのクローン作成にも使用できます。	FileSnap
SFHA Solutions の SmartTier を使用して、経過時間、優先度、アクセス率の基準に基づいてストレージ階層にデータを移動することにより、ストレージ利用率を最大化します。 p.247 の「SmartTier について」を参照してください。	SmartTier
物理的な共有ストレージを使わずにデータ冗長性、高可用性、ディザスタリカバリを実現するために、ストレージの利用率を最大化します。	Flexible Storage Sharing
高度で、カスタマイズ可能なヒューリスティックを使って、I/O キャッシュで SSD（ソリッドステートドライブ）でのデータ効率を高め、キャッシュするデータと、そのデータをキャッシュから削除する方法を特定します。	VxVM ボリュームで実行しているアプリケーションの SmartIO 読み取りキャッシュ VxFS ファイルシステムで実行しているアプリケーションの SmartIO 読み取りキャッシュ VxFS ファイルシステムで実行しているアプリケーションの SmartIO 書き込みキャッシュ VxFS ファイルシステムのデータベースの SmartIO キャッシュ VxVM ボリュームのデータベースの SmartIO キャッシュ データベースの SmartIO ライトバックキャッシュは SFRAC ではサポートされない 『Veritas InfoScale 7.4.2 SmartIO for Solid-State Drives ソリューションガイド』を参照してください。
vSphere 環境で計画されたフェールオーバーを実行する仮想マシンの保守や、計画外のエラー発生時における Just In Time Availability ソリューションを使用するアプリケーションのリカバリを計画します。	Just In Time Availability ソリューション

使用例	Veritas InfoScale の機能
高度な形式またはストレージ環境の 4 K (4096 バイト) セクタデバイス (4 KB でフォーマット) でサポートを提供する Veritas InfoScale ソリューションを使用して、ストレージデバイスのネーティブ形式と最適化形式を改善します。	Veritas InfoScale 4 K セクタのデバイスサポートのソリューション
ETL パイプラインなど、柔軟なデータの共有を必要とするデータウェアハウス内の複数のパラレルアプリケーションでは、ある段階の出力が次の段階の入力になります。(たとえば、会計システムでは販売、給与、購入などさまざまなアプリケーションからのデータを結合する必要があります)	Verita InfoScale のアプリケーション分離 関連情報: ディスクグループのサブクラスタ化を使用した CVM 環境でのアプリケーションの分離 CVM 環境でのアプリケーション分離機能の有効化 CVM クラスタでのアプリケーション分離機能の無効化 例: マスターフェールオーバーへのサブクラスタノードの優先設定値の設定 手動でのディスクグループマスターの変更 詳しくは、『Storage Foundation Cluster File System High Availability 管理者ガイド』を参照してください。
SAN ストレージからすべての CVM ノードへの完全なゾーン化の要件を緩和します。これにより、独立したクラスタの統合が可能となり、管理が容易になります。	Verita InfoScale のアプリケーション分離 関連情報: ディスクグループのサブクラスタ化を使用した CVM 環境でのアプリケーションの分離 CVM 環境でのアプリケーション分離機能の有効化 CVM クラスタでのアプリケーション分離機能の無効化 例: マスターフェールオーバーへのサブクラスタノードの優先設定値の設定 手動でのディスクグループマスターの変更 詳しくは、『Storage Foundation Cluster File System High Availability 管理者ガイド』を参照してください。

使用例	Veritas InfoScale の機能
複数の独立したクラスタ化したアプリケーションを有効にして、拡張性のある DAS ストレージの共有プールを使用します。これは、ストレージ専用ノードの追加を促進し、増加するストレージ容量に対してクラスタ化を行い、専用アプリケーションを使用するノードを計算します。	Verita InfoScale のアプリケーション分離 関連情報: ディスクグループのサブクラスタ化を使用した CVM 環境でのアプリケーションの分離 CVM 環境でのアプリケーション分離機能の有効化 CVM クラスタでのアプリケーション分離機能の無効化 例: マスターフェールオーバーへのサブクラスタノードの優先設定値の設定 手動でのディスクグループマスターの変更 詳しくは、『Storage Foundation Cluster File System High Availability 管理者ガイド』を参照してください。

2

Veritas InfoScale 製品を使用した Oracle の配備

- 第2章 Storage Foundation 環境への Oracle オプションの配備
- 第3章 Storage Foundation を使用した Oracle の配備
- 第4章 Storage Foundation を使用したオフホスト設定での Oracle の配備
- 第5章 High Availability を使用した Oracle の配備
- 第6章 ディザスタリカバリ用 VVR (Volume Replicator) を使用した Oracle の配備

2

Storage Foundation 環境への Oracle オプションの配備

この章では以下の項目について説明しています。

- Storage Foundation 環境の Oracle の配備オプション
- Storage Foundation 環境での Oracle 単一インスタンス
- Storage Foundation 環境でオフホストを使う単一インスタンスの Oracle
- Storage Foundation High Availability を用いた高可用性クラスタの単一インスタンス Oracle
- SF Cluster File System HA 環境の並列クラスタの Oracle の単一インスタンス
- Storage Foundation for Oracle RAC 環境の Oracle RAC について
- レプリケーションされた Storage Foundation 環境での Oracle について
- 仮想化環境での Oracle と Storage Foundation の配備
- Storage Foundation SmartMove およびシンプロビジョニングによる Oracle の配備

Storage Foundation 環境の Oracle の配備オプション

次の設定で Veritas InfoScale Storage Foundation and High Availability 製品を備えた Oracle を配備できます。

- Storage Foundation 環境での Oracle 単一インスタンス

- Storage Foundation 環境での Oracle 単一インスタンスとオフホスト機能付き Oracle RAC
- Oracle 単一インスタンスと高可用性（HA）
- Oracle RAC と Storage Foundation for Oracle RAC
- Oracle 単一インスタンスと SFCFSHA
- Oracle 単一インスタンスと Veritas Volume Replicator（VVR）または障害回復用の Oracle Data Guard
- 仮想環境での Oracle 単一インスタンスと Storage Foundation

SFDB（Storage Foundation for Database）ツールは、これらのセットアップをすべてサポートします。

Storage Foundation 環境での Oracle 単一インスタンス

Storage Foundation（SF）で単一インスタンスの Oracle を配備する場合、セットアップ設定は次の条件を反映します。

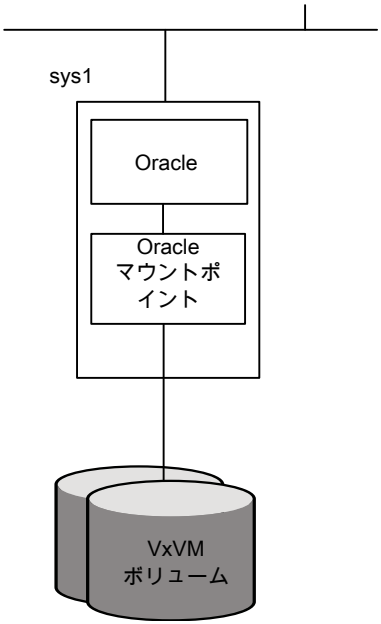
- Oracle の単一インスタンスは SF のある *sys1* に設定されます。
- データベースとデータファイルは *sys1* でオンラインです。
- *sys1* 上で SFDB ツールコマンドを実行する必要があります。

Storage Foundation for Databases（SFDB）リポジトリまたはリポジトリデータベースについて詳しくは次を参照してください。

p.64 の「SFDB (Storage Foundation for Databases)リポジトリについて」を参照してください。

「図 2-1」に、Storage Foundation 環境での Oracle の単一インスタンスの配備について示します。

図 2-1 Storage Foundation での Oracle の単一インスタンス



Storage Foundation 環境でオフホストを使う単一インスタンスの Oracle

オフホスト設定で Storage Foundation を使う単一インスタンスの Oracle を配備する場合、設定は次の条件を反映します。

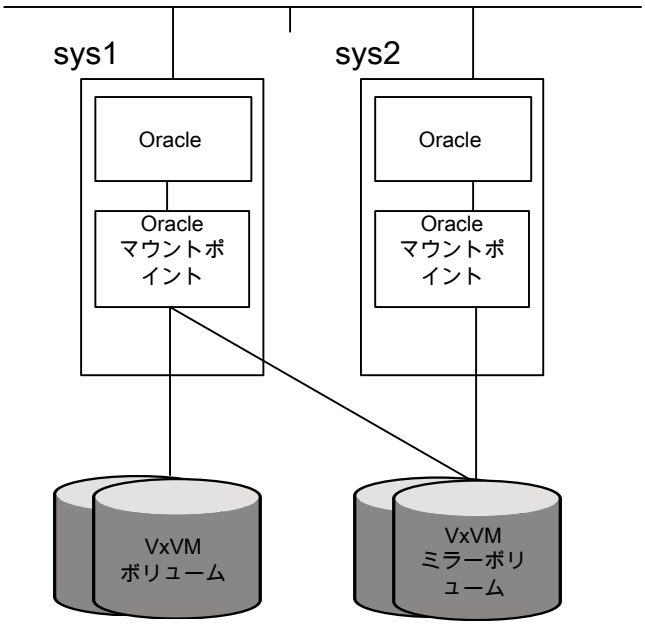
- Oracle の単一インスタンスは SF のある *sys1* に設定されます。
- データベースとデータファイルは *sys1* でオンラインです。
- *sys1* と *sys2* は同じストレージを共有します。

Storage Foundation for Databases（SFDB）リポジトリまたはリポジトリデータベースについて詳しくは次を参照してください。

p.64 の「SFDB (Storage Foundation for Databases)リポジトリについて」を参照してください。

「図 2-2」は、Storage Foundation 環境でオフホストを使う単一インスタンスの Oracle の配備を示しています。

図 2-2 Storage Foundation 環境でオフホストを使う単一インスタンスの Oracle



Storage Foundation High Availability を用いた高可用性クラスタの単一インスタンス Oracle

SFHA（Storage Foundation High Availability）で Oracle を配備する場合、セットアップ設定には次の条件が反映されます。

- Oracle の高可用性単一インスタンスは SFHA のある *sys1* と *sys2* に設定されます。

- データベースとデータファイルは *sys1* でオンラインです。
- データベースがオンラインである *sys1* で SFDB ツールコマンドを実行する必要があります。

Storage Foundation for Databases（SFDB）リポジトリまたはリポジトリデータベースについて詳しくは次を参照してください。

p.64 の「SFDB (Storage Foundation for Databases)リポジトリについて」を参照してください。

図 2-3 および 図 2-4 に、Storage Foundation High Availability 環境での Oracle 単一インスタンス、およびそのようなセットアップでのフェールオーバーシナリオを示します。

図 2-3 Oracle 単一インスタンスと SFHA

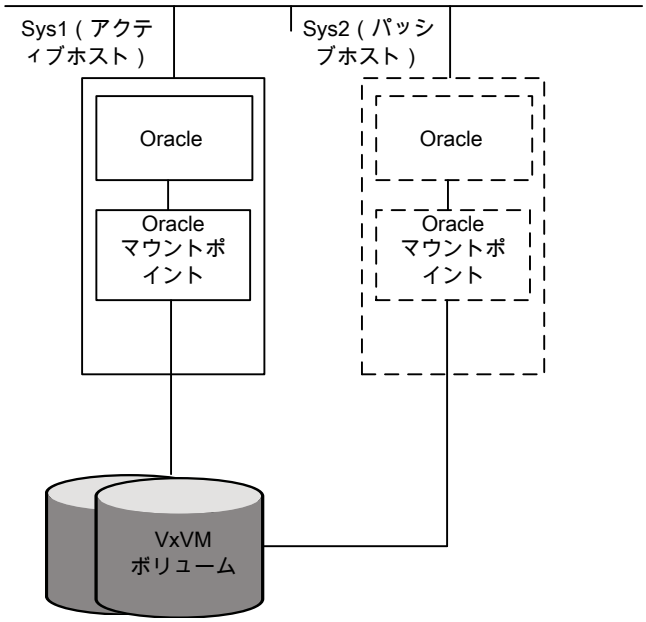
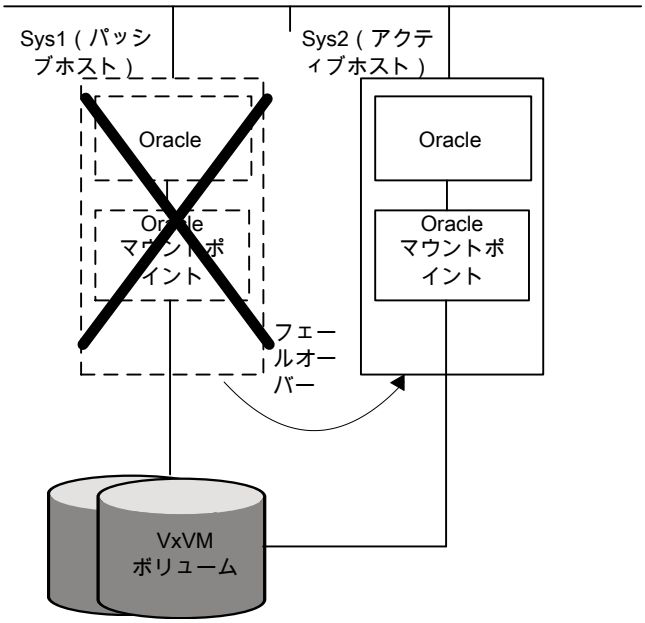


図 2-4　Oracle 単一インスタンスと SFHA フェールオーバー設定



SF Cluster File System HA 環境の並列クラスタの Oracle の単一インスタンス

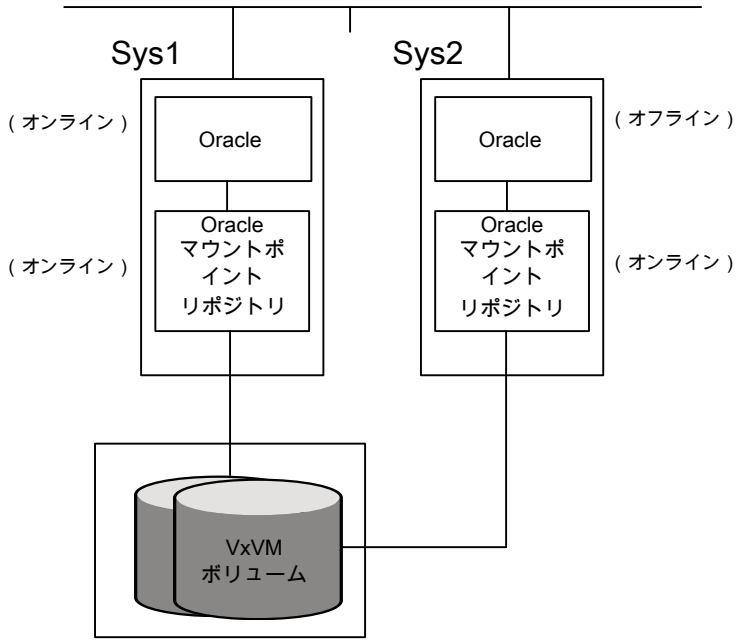
SF Cluster File System HA で Oracle の単一インスタンスを配備する場合、セットアップ設定は次の条件を反映します。

- Oracle を使用する高可用性並列クラスタは、SF Cluster File System HA のある *sys1* と *sys2* に設定されます。
- データベースは *sys1* でオンラインです。
- データファイルは *sys1* と *sys2* にマウントされ、共有されます。
- データベースリポジトリは *syst1* と *sys2* にマウントされ、共有されます。

- *sys2* では SFDB ツールコマンドは失敗します。

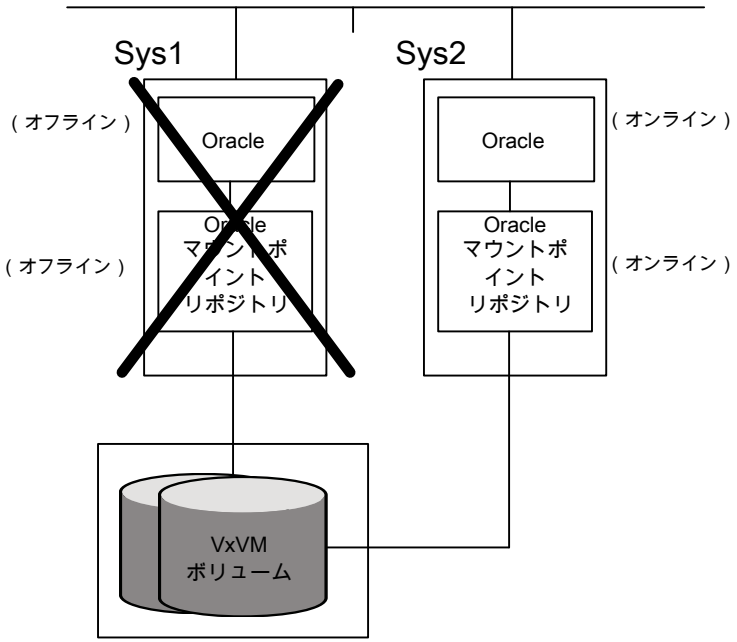
次の図では、リポジトリディレクトリは Oracle マウントポイントに存在します。

図 2-5 SF Cluster File System HA の設定例



SF Cluster File System HA では、バックアップシステムへのフェールオーバーは手動ではなく自動で行われます。

図 **2-6** SFCFS HA フェールオーバーの例



Storage Foundation for Oracle RAC 環境の Oracle RAC について

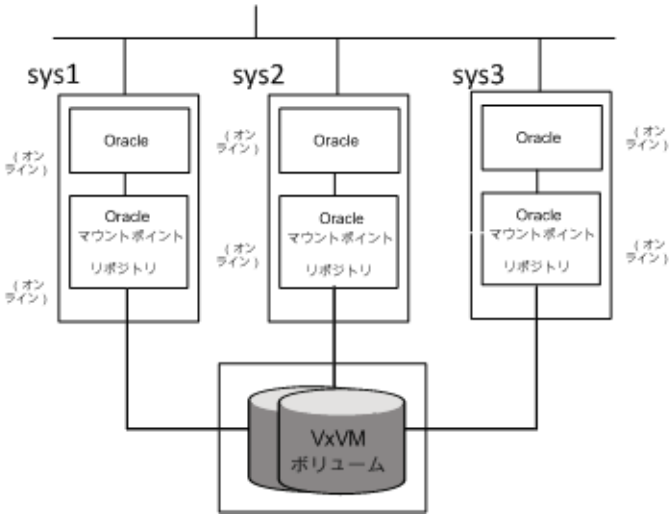
Storage Foundation for Oracle RAC で Oracle RAC を配備する場合、セットアップ設定は次の条件を反映します。

- Oracle の複数インスタンスを使用する高可用性並列クラスタは、SF for Oracle RAC のある *sys1*、*sys2*、*sys3* に設定されます。
- データベースは *sys1*、*sys2*、*sys3* でオンラインです。
- データファイルは *sys1*、*sys2*、*sys3* にマウントされ、共有されます。
- SFDB ツールは *sys1*、*sys2*、*sys3* にマウントされ、共有されます。

- この設定ではクラスタ化された ODM がサポートされています。

次の図では、リポジトリディレクトリは Oracle マウントポイントに存在します。

図 2-7 Storage Foundation for Oracle RAC 環境の Oracle RAC



SF Oracle RAC 構成の場合、システムは並行してオンラインとなり、クラスタ内でフェールオーバー機構を使いません。

『Veritas InfoScale Storage Foundation for Oracle RAC 管理者ガイド』を参照してください。

レプリケーションされた Storage Foundation 環境での Oracle について

Oracle 環境では、Veritas Volume Replicator（VVR）と Oracle Data Guard の 2 つのレプリケーションオプションがサポートされています。

SFDB（Storage Foundation for Database）ツールは、セカンダリサイトにおいて、単一インスタンスモードのこれらのレプリケーションツールをサポートしています。

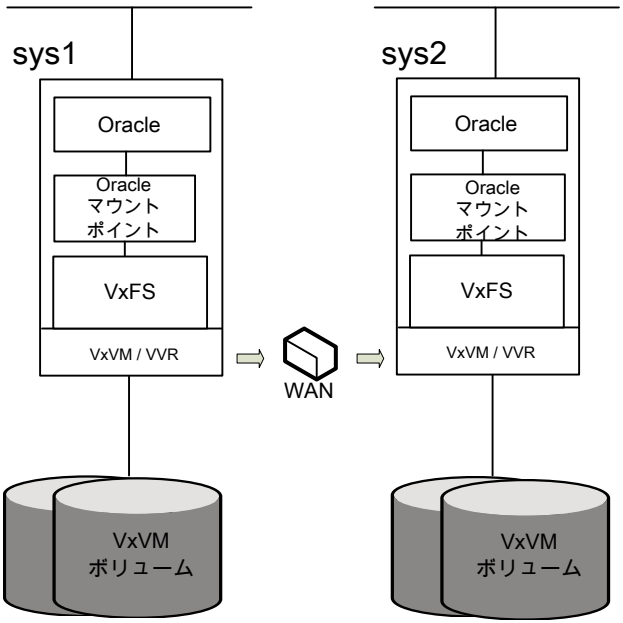
Storage Foundation 環境にディザスタリカバリ用の VVR を使用して Oracle の単一インスタンスを配備する場合、セットアップ設定は次の条件を反映します。

- Oracle の単一インスタンスは *sys1* で設定されます。
- データベースとデータファイルは *sys1* でオンラインです。
- `ORACLE_HOME` は *sys2* で設定されます。

- *sys1* のデータボリュームは *sys2* に対するレプリケーションに設定されます。
- *sys1* ボリュームのアプリケーションデータの一貫性のあるコピーは *sys2* で利用可能です。

図 2-8 に、VVR が有効な Storage Foundation 環境での Oracle 単一インスタンスの配備を示します。

図 2-8 Oracle の単一インスタンス - VVR を使用した Storage Foundation



仮想化環境での Oracle と Storage Foundation の配備

仮想コンピュータ環境で SFDB（Storage Foundation for Databases）ツールを配備する場合、次がサポートされています。

- AIX での WPAR

- Linux での VMware
- Solaris での Oracle VM Server for SPARC

Storage Foundation SmartMove およびシンプロビジョニングによる Oracle の配備

Storage Foundation 製品と Oracle データベースで SmartMove とシンプロビジョニングを使うことができます。

データファイルが削除された場合、下位のデバイスがシン再生 LUN の場合に、これらのファイルによって使われたストレージ領域を再生できます。このためには Storage Foundation シン再生機能を使用します。

詳しくは、『Storage Foundation 管理者ガイド』を参照してください。

3

Storage Foundation を使用した Oracle の配備

この章では以下の項目について説明しています。

- Oracle データベースを配備するためのタスク
- Oracle のストレージの計画
- Oracle の配備のためのボリュームレイアウトの選択について
- Oracle を配備するためのディスクグループの設定
- Oracle を配備するためのボリュームの作成
- Oracle を配備するための VxFS ファイルシステムの作成
- Oracle を配備するためのファイルシステムのマウント
- Oracle のインストールとデータベースの作成

Oracle データベースを配備するためのタスク

Storage Foundation 環境で Oracle 単一インスタンスのデータベースを配備する場合は、下記に挙げた順序でタスクを完了させます。

作成するデータベースに必要なファイルシステムの数とサイズを決定する。	p.46 の「Oracle のストレージの計画」を参照してください。

ボリュームレイアウトを作成する。	p.47 の「Oracle の配備のためのボリュームレイアウトの選択について」を参照してください。 p.47 の「Oracle を配備するためのディスクグループの設定」を参照してください。 p.49 の「Oracle を配備するためのディスクグループ設定のガイドライン」を参照してください。 p.49 の「Oracle を配備するためのボリュームの作成」を参照してください。 p.50 の「Oracle を配備するためのボリューム設定のガイドライン」を参照してください。
ファイルシステムを作成してマウントする。	p.50 の「Oracle を配備するための VxFS ファイルシステムの作成」を参照してください。 p.51 の「Oracle を配備するためのファイルシステムの作成ガイドライン」を参照してください。 p.52 の「Oracle を配備するためのファイルシステムのマウント」を参照してください。
Oracle をインストールしてデータベースを作成する。	p.53 の「Oracle のインストールとデータベースの作成」を参照してください。

Oracle のストレージの計画

データベースのサイズに応じてファイルシステムの数および各ファイルシステムのサイズを決定します。Oracle データベースで次のファイルシステムを作成することをお勧めします。

- Oracle のホーム用ファイルシステム
  このファイルシステムには Oracle バイナリがインストールされます。
- データファイル用ファイルシステム
  このファイルシステムにはデータファイルが格納されます。ファイルシステムのサイズはデータベースのサイズで決まります。ファイルシステムのニーズを満たすようにボリュームを作成する必要があります。ディスクのミラー化を使うとディスク障害に備えることができ、ストライプ化を使うとパフォーマンスを向上できます。
  データベースの種類、I/O パターン、インデックス、一時ファイルなどの要素に応じて、データファイルの格納に単一または複数のファイルシステムを作成できます。
- REDO ログ用ファイルシステム
  このファイルシステムには REDO ログが格納されます。よりよいパフォーマンスを得るために、REDO ログ用のファイルシステムは高速ストレージデバイスに作成することをお勧めします。

- アーカイブ REDO ログ用ファイルシステム
  このファイルシステムにはアーカイブ REDO ログが格納されます。

Oracle の配備のためのボリュームレイアウトの選択について

VxVM（Veritas Volume Manager）ではさまざまなレイアウトを設定することができ、これによりユーザーは、データベースのパフォーマンスと可用性の必要条件に合わせてデータベースを設定できます。ボリュームレイアウトを適切に選択することによって、データベースの作業負荷に対して最適なパフォーマンスがもたらされます。

データベースのパフォーマンスに影響を与える 1 つの要因にディスク上の表領域の配置があります。

ディスク I/O は、データベースのパフォーマンスを決める重要な要素の 1 つです。I/O 負荷が分散されると通常は最適なパフォーマンスが得られます。分散型 I/O が実現されるようにデータベースオブジェクトのレイアウトを設計することは、データベースを設定するうえで重要な一歩となります。表領域を配置する場所を決めるときに、今後の使用パターンの予測を立てることが難しい場合がよくあります。VxVM では、初回のデータベースの設定や、ニーズの変化に合わせた継続的なデータベースパフォーマンスの向上のために、ストレージを柔軟に設定できます。また、ボリュームを複数デバイスに分割してデータ配置の精度を高めることができます。ストライプボリュームを使用することで、、I/O は複数のディスクドライブに渡って分散できます。ほとんどのデータベースでは、データベースに応じて、異なるコンテナであれ表領域であれ利用可能なディスク間に配布されるのを確認すれば十分である可能性があります。

ストライプ化はまた順次テーブルスキャンのパフォーマンスの向上に役立ちます。テーブルを複数のデバイスにわたってストライプ化する場合は、Oracle のパラメータ `DB_FILE_MULTIBLOCK_READ_COUNT` を、フルストライプサイズを `DB_BLOCK_SIZE` で除算した値の倍数に設定することによって、高転送帯域幅を確保できます。

Oracle データベースで Database FlashSnap 機能（ポイントインタイムコピー）を使用し、それを同じホストで、またはオフホスト処理かバックアップで使用する計画の場合は、ボリュームのレイアウトが FlashSnap の必要条件を満たすようにする必要があります。

Oracle を配備するためのディスクグループの設定

データベースのボリュームとファイルシステムを作成する前に、各データベースのディスクグループをセットアップする必要があります。

ディスクグループを作成する前に、ディスクグループ設定のガイドラインを確認します。

p.49 の「Oracle を配備するためのディスクグループ設定のガイドライン」を参照してください。

ディスクグループを作成するには

◆ 次のようにして vxdg コマンドを使用します。

```
# /opt/VRTS/bin/vxdg init disk_group disk_name=disk_device
```

たとえば、raw ディスクパーティションで *PRODdg* という名前のディスクグループを作成する場合、ディスク名 *PRODdg01* はディスクグループ内のディスクを参照します。

AIX

```
# /opt/VRTS/bin/vxdg init PRODdg PRODdg01=Disk_0
```

Linux

```
# /opt/VRTS/bin/vxdg init PRODdg PRODdg01=sda
```

Solaris

```
# /opt/VRTS/bin/vxdg init PRODdg PRODdg01=c1t1d0s2
```

ディスクグループにディスクを追加するには

◆ 次のようにして vxdg コマンドを使用します。

```
# /opt/VRTS/bin/vxdg -g disk_group adddisk disk_name=disk_device
```

たとえば、PRODdg02 という名前のディスクをディスクグループ PRODdg に追加するには次のようにします。

AIX

```
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg02=Disk_0

# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg03=Disk_1

# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg04=Disk_2
```

Linux

```
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg02=sda

# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg03=sdb

# /opt/VRTS/bin/vxdg -g PRODdg adddisk  PRODdg04=sdc
```

Solaris

```
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg02=c1t2d0s2
```

```
# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg03=c1t3d0s2

# /opt/VRTS/bin/vxdg -g PRODdg adddisk PRODdg04=c1t4d0s2
```

Oracle を配備するためのディスクグループ設定のガイドライン

ディスクグループを設定する場合は次のガイドラインに従います。

- 新しいディスクグループを作成するために使用できるのは、オンライン状態であり、どのディスクグループにも属していないディスクのみです。
- 各データベースに対して 1 つのディスクを作成します。
- ディスクグループ名は一意である必要があります。環境変数 ORACLE_SID と接尾辞 dg で指定される Oracle データベースインスタンスを使って各ディスクグループに名前を付けます。接尾辞 dg は、オブジェクトをディスクグループとして識別するのに役立ちます。
- 各ディスク名はディスクグループ内で一意である必要があります。
- VVR 環境では、同じディスクグループにないファイルシステムやボリュームを使ってデータベースファイルを作成しないでください。

メモ: VxVM コマンドに関連するディスクグループを実行するにはいずれの場合にも root 権限を持っている必要があります。

詳しくは、『Storage Foundation 管理者ガイド』を参照してください。

Oracle を配備するためのボリュームの作成

VxVM（Veritas Volume Manager）は、論理ボリュームを使用してディスク領域を編成して管理します。ボリュームは、1 つ以上の物理ディスクの部分によって構成されるため、物理ディスクの制約を受けません。

ボリュームを作成する前に、ボリュームの作成ガイドラインを確認します。

p.50 の「Oracle を配備するためのボリューム設定のガイドライン」を参照してください。

ボリュームを作成するには

◆ 次のように、vxassist コマンドを使います。

```
# /opt/VRTS/bin/vxassist -g disk_group make volume_name
volume_size
 disk_name
```

次は vxassist コマンドを使用してボリュームを作成する例です。

次の例では、PRODdg ディスクグループ上に db01 という 1 GB のボリュームを作成します。

```
#/opt/VRTS/bin/vxassist -g PRODdg make db01 1g PRODdg01
```

Oracle を配備するためのボリューム設定のガイドライン

ボリュームレイアウトを選定する場合は次のガイドラインに従います。

- ストライプ化およびミラー化された（RAID-0+1）ボリューム上に作成されたファイルシステムのデータベースログファイルは、インデックスまたは表領域とは別に置きます。必要に応じて複数のデバイスをストライプ化してより大きいボリュームを作成してください。信頼性を向上させるためにミラー化を使用します。REDO ログに VxVM RAID-5 を使用しないでください。
- 通常のシステム可用性が許容される場合は、ほとんどの OLTP 作業負荷ではストライプボリューム上に作成されたファイルシステムに表領域を置きます。
- ストライプボリュームは最低でも 4 つのディスク間で作成します。ディスクコントローラをまたがるストライプ化を試行します。
  順次スキャンでは、多過ぎるディスクやコントローラ間でのストライプ化は避けてください。単一スレッドで順次スキャンを処理する場合、ディスクの速度についていけなくなることがあります。
- ほとんどの作業負荷では、ストライプ化されたボリュームについてデフォルトの 64 K ストライプユニットのサイズを使います。
- システム可用性が重要な場合、書き込みを集中的に行うほとんどの OLTP 作業負荷ではミラー化を使います。DRL（dirty region logging）をオンにして、システムがクラッシュした場合に備えて高速ボリューム再同期を可能にします。
- ほとんどの DSS（意思決定支援システム）作業負荷では、順次スキャンが一般的であり、さまざまなストライプ化戦略やストライプユニットのサイズを試してみます。最も頻繁にアクセスするテーブルまたは一緒にアクセスするテーブルを別のストライプボリュームに置いてデータ転送の帯域幅を向上させます。

Oracle を配備するための VxFS ファイルシステムの作成

VxFS（Veritas File System）ファイルシステムを作成するには、`mkfs` コマンドまたは `mkfs_vxfs` コマンドを使います。

VxFS ファイルシステムを作成する前にファイルシステムの作成のガイドラインを確認します。

p.51 の「Oracle を配備するためのファイルシステムの作成ガイドライン」を参照してください。

既存のボリュームで VxFS ファイルシステムを作成するには

◆ 次のように、mkfs コマンドを使います。

- AIX

```
# /usr/sbin/mkfs -V vxfs generic_options¥
 -o specific_options special size
```

- Solaris

```
# /usr/sbin/mkfs -F vxfs generic_options¥
 -o specific_options special size
```

- Linux

```
# /usr/sbin/mkfs -t vxfs generic_options¥
 -o specific_options special size
```

ここで、

- vxfs はファイルシステムのタイプです
- generic_options はほとんどのファイルシステムに共通のオプションです
- specific_options は VxFS ファイルシステム固有のオプションです
- special はファイルシステムを作成する RAW キャラクタデバイスまたは VxVM ボリュームの絶対パス名です
- （オプション）size は新しいファイルシステムのサイズです

サイズを指定しない場合、ファイルシステムは下位ボリュームと同じ規模になります。

たとえば、ブロックサイズが 8 KB で新しく作成される db01 ボリュームで 2 GB より大きいファイルをサポートする VxFS ファイルシステムを作成するには、次のコマンドを使います。

- `# /usr/sbin/mkfs -V vxfs -o largefiles,bsize=8192,logsize=2000 ¥ /dev/vx/rdsk/PRODdg/db01`

-o largefiles オプションにより、2 GB より大きいファイルを作成できます。

メモ: この例ではサイズが指定されていないため、ファイルシステムのサイズは自動的に計算されてファイルシステムが作成されるボリュームと同じサイズになります。

Oracle を配備するためのファイルシステムの作成ガイドライン

VxFS ファイルシステムを作成する場合、次のガイドラインに従ってください。

- データベースのファイルシステムを作成する場合、最大ブロックサイズとログサイズを指定します。
- ファイルシステムのインテントログ機能は無効にしません。
- REDO ログ、制御ファイル、データファイル、tmp ファイル、アーカイブ REDO ログについては、別々のファイルシステムを作成します。
- コマンドラインを使う場合、マウントポイントを使って下位ボリュームの名前を付けます。たとえば、ミラー化されたボリュームで /db01 という名前のファイルシステムを作成する場合、ボリュームには db01 という名前を付け、ミラーには db01-01 と db01-02 という名前を付けて、設定オブジェクトに関連させます。vxassist コマンドまたは GUI を使っている場合、これは透過的です。
- Oracle データベースのブロックサイズはファイルシステムブロックサイズの倍数になる必要があります。可能であれば、同じサイズに保ちます。

詳しくは、『Storage Foundation 管理者ガイド』を参照してください。

Oracle を配備するためのファイルシステムのマウント

VxFS ファイルシステムを作成した後で、root ユーザーとして、mount コマンドを使ってファイルシステムをマウントします。

詳しくは、mount コマンドと mount_vxfs コマンドのマニュアルページを参照してください。

ファイルシステムをマウントするには

◆ 次のようにして mount コマンドを使用します。

- AIX

```
# /usr/sbin/mount -V vxfs special /mount_point
```

- Solaris

```
# /usr/sbin/mount -F vxfs special /mount_point
```

- Linux

```
# /usr/sbin/mount -t vxfs special /mount_point
```

ここで、

- vxfs はファイルシステムのタイプです
- special は、ブロックの特殊デバイスです。
- /mount_point は、ファイルシステムがマウントされるディレクトリです。

たとえば、大容量ファイルをサポートする /db01 という名前のファイルシステムをボリューム /dev/vx/dsk/PRODdg/db01 にマウントするには次のようにします。

- ```
  # /usr/sbin/mount -V vxfs -o largefiles /dev/vx/dsk/PRODdg/db01 ¥
  /db01
  ```

Oracle のインストールとデータベースの作成

runInstaller や dbca などのツールを使って Oracle をインストールしてデータベースを作成します。Oracle をインストールしてデータベースを作成する過程では、特定の目的のために作成されたさまざまなファイルシステムを使用します。

p.46 の「Oracle のストレージの計画」を参照してください。

Oracle を配備するためのデータベースレイアウトの注意事項とサポートされている設定を確認します。

p.183 の「データベースレイアウトの注意事項」を参照してください。

p.184 の「サポート対象の Oracle 設定」を参照してください。

Oracle ソフトウェアのインストールと Oracle データベースの作成について詳しくは、Oracle 社のマニュアルを参照してください。
  ```

4

Storage Foundation を使用したオフホスト設定での Oracle の配備

この章では以下の項目について説明しています。

- データベースのオフホスト設定の必要条件

データベースのオフホスト設定の必要条件

オフホスト構成で SFDB（Storage Foundation Database）を使用して Oracle 単一インスタンスデータベースを設定する場合は、次の点を確認してください。

- Veritas InfoScale Storage Foundation 環境に単一インスタンス Oracle データベースを配備するためのタスクがすべて完了している。
  p.45 の「Oracle データベースを配備するためのタスク」を参照してください。
- 次の必要条件が満たされていること。
  - ファイルはすべて VxVM ボリュームにわたる VxFS ファイルシステム上に配置されている。RAW デバイスはサポート対象外となっている。
  - データベースファイルへのシンボリックリンクが存在しない。
  - ORACLE_HOME が別のファイルシステム上にある。
  - アーカイブログは、それぞれ別の VxFS ファイルシステム上に配置され、さらに Oracle データファイルまたは ORACLE_HOME を含む VxFS ファイルシステムとは離れている。
  - データベースに BFILES と外部表が格納されていない。

- Oracle データファイル、アーカイブログ、REDO ログ、制御ファイルは、1 つまたは複数のディスクグループに格納されている。
- フラッシュ回復領域が ORACLE_HOME とは別のファイルシステムに存在する。
- プライマリホストとセカンダリホストにインストールされている製品のバージョンが同じである。
- 同じバージョンの Oracle が両方のホストにインストールされており、Oracle のバイナリとデータファイルが異なるボリュームとディスク上に存在する。
- データベースユーザーとグループの UNIX ログインが両方のホストで同じである。UNIX UID と GID も同じである必要があります。
- 両方のホストに Veritas InfoScale Enterprise ライセンスが付与されている必要があります。

5

High Availability を使用した Oracle の配備

この章では以下の項目について説明しています。

- Oracle を HA 設定に配備するためのタスク
- データベースを高可用性にする VCS 設定

Oracle を HA 設定に配備するためのタスク

Oracle 単一インスタンスデータベースを Storage Foundation High Availability（SFHA）環境に配備する場合は、次のタスクを完了させます。

Oracle 単一インスタンスデータベースを Storage Foundation の環境に配備するためのタスクを完了させる。	p.45 の「Oracle データベースを配備するためのタスク」を参照してください。
VCS を設定してデータベースを高可用性にする。	p.56 の「データベースを高可用性にする VCS 設定」を参照してください。

データベースを高可用性にする VCS 設定

Oracle データベースの可用性を高めるには、データベース設定を VCS（Cluster Server）によって制御する必要があります。

『Veritas InfoScale Cluster Server 管理者ガイド』を参照してください。

6

ディザスタリカバリ用 VVR (Volume Replicator) を使用した Oracle の配備

この章では以下の項目について説明しています。

- ディザスタリカバリ用の VVR を備えた Oracle の配備について
- ディザスタリカバリ用の VVR を備えた Oracle を配備するタスク
- Oracle を配備するためのプライマリノードの設定
- Oracle を配備するためのセカンダリノードの設定
- Oracle を配備するためのプライマリとセカンダリのノード間の VVR の設定

ディザスタリカバリ用の VVR を備えた Oracle の配備について

VVR（Veritas Volume Replicator）は、Oracle 環境で効果的なディザスタリカバリ計画に役立つよう設計されたデータレプリケーションソフトウェアです。VVR を使うことにより、1 つ以上のリモートサイトで一貫性のある Oracle データのコピーを保持できます。VVR を使うと、レプリケーション元のボリュームにアプリケーションが書き込んだデータを、距離に関係なく 1 つ以上のリモートサイトにレプリケートすることが可能となります。レプリケーション元で災害が発生した場合には、リモートの場所にある Oracle データのコピーを使い、リモートの場所でアプリケーションを再起動できます。

『Veritas InfoScale 7.4.2 レプリケーション管理者ガイド』を参照してください。

ディザスタリカバリ用の VVR を備えた Oracle を配備するタスク

障害回復用の Veritas Volume Replicator（VVR）が有効な Storage Foundation 環境で Oracle データベースを設定する場合は、次のタスクを順番に実行します。

VVR 環境にプライマリノードを設定する。	p.58 の「Oracle を配備するためのプライマリノードの設定」を参照してください。
VVR 環境にセカンダリノードを設定する。	p.59 の「Oracle を配備するためのセカンダリノードの設定」を参照してください。
プライマリノードとセカンダリノード間で VVR を設定する。	p.61 の「Oracle を配備するためのプライマリとセカンダリのノード間の VVR の設定」を参照してください。

Oracle を配備するためのプライマリノードの設定

Linux プラットフォームの場合の手順例を次に示します。

プライマリノードを設定するには

1. プライマリノードにディスクグループを作成します。

```
# vxdg init vvrdg sdag sdah sdai sdaj
```

2. プライマリノードにデータボリュームを作成します。

```
# vxassist -g vvrdg make datavol 4500M sdag
```

3. プライマリノードにアーカイブログボリュームを作成します。

```
# vxassist -g vvrdg make archvol 4500M sdah
```

4. データボリュームとアーカイブログボリュームにファイルシステムを作成しマウントします。

```
# mkfs -t vxfs /dev/vx/rdsk/vvrdg/datavol
# mkdir /data
# mount -t vxfs /dev/vx/dsk/vvrdg/datavol /data
# mkfs -t vxfs /dev/vx/rdsk/vvrdg/archvol
# mkdir /arch
# mount -t vxfs /dev/vx/dsk/vvrdg/archvol /arch
```

5 データボリュームとアーカイブログボリュームの所有権を Oracle ユーザーに割り当てます。

```
# chown ora11gr2:dba /data /arch
```

6 データボリュームとアーカイブログボリュームのみを持つディスク上の各ボリュームの SRL（Storage Replicator Log）を作成します。

```
# vxassist -g vvrdg make datavol_srl 4500M sdai
# vxassist -g vvrdg make archvol_srl 4500M sdaj
```

Oracle を配備するためのセカンダリノードの設定

セカンダリノードを設定するには

1 プライマリノードのディスクグループと同じ名前のディスクグループを作成します。

```
# vxdg init vvrdg sdag sdah sdai sdaj
```

2 データボリュームとアーカイブログボリュームのディレクトリを作成し、Oracle ユーザーに所有権を割り当てます。

```
# mkdir -p /data
# mkdir -p /arch
# chown -R ora11gr2:dba /data
# chown -R ora11gr2:dba /arch
```

3 プライマリのノードボリュームと同じ名前と長さのデータボリュームとアーカイブログボリュームを作成します。

```
# vxassist -g vvrdg make datavol 4500m alloc=sdj,sdk
# vxassist -g vvrdg make archvol 4500m alloc=sdl,sdm
```

4 データボリュームとアーカイブログボリュームに DCM（Data Change Map）のログを追加します。

```
# vxassist -g vvrdg addlog datavol logtype=dcm nlog=2
logdisk=sdj,sdk
# vxassist -g vvrdg addlog archvol logtype=dcm nlog=2
logdisk=sdl,sdm
```

5 データボリュームとアーカイブログボリュームが含まれるディレクトリの所有権と権限を変更します。

```
# chown -R ora11gr2:dba /data*
# chown -R ora11gr2:dba /arch*
```

6 DRL（Dirty Region Logging）が無効の状態でスナップショット操作を行うためのデータボリュームとアーカイブログボリュームを準備します。

```
# vxsnap -g vvrdg prepare datavol drl=no alloc=sdn,sdo
# vxsnap -g vvrdg prepare archvol drl=no alloc=sdp,sdq
```

メモ: セカンダリホスト上のデータを使ってオフホスト処理の操作を実行するには、セカンダリデータボリュームのスナップショットを使います。読み取り専用モードの場合でも、セカンダリ RVG ボリュームを直接マウントしないでください。

スナップショットは特定時点におけるデータボリュームのイメージです。データボリュームはレプリケーションにより更新されますが、スナップショットのデータはある時点のデータボリュームのコピーとして、様々な目的に使用することが可能です。

---

7 データボリュームとアーカイブログボリュームがインスタントスナップショット用に準備されているかどうかを検証します。

```
# vxprint -g vvrdg -F%instant datavol
# vxprint -g vvrdg -F%instant archvol
```

メモ: VVR はインスタントスナップショットと従来のスナップショットの 2 種類のスナップショットの両方を使うことが可能です。VxVM のインスタントスナップショット機能には、Veritas InfoScale Enterprise および Veritas InfoScale Storage のライセンスが必要です。

---

8 サードミラーブレークオフスナップショットの作成に使用するために、データボリュームとアーカイブログボリュームにミラーを追加します。

```
# vxsnap -g vvrdg addmir datavol alloc=sdr
# vxsnap -g vvrdg addmir archvol alloc=sds
```

9 データボリュームとアーカイブログボリュームのブレークオフデータプレックスの dbed_flashsnap タグを設定します。

```
# vxedit -g vvrdg set putil2=dbed_flashsnap datavol-04
# vxedit -g vvrdg set putil2=dbed_flashsnap archvol-04
```

10 セカンダリのノードのデータボリュームとアーカイブログボリュームの SRL（Storage Replicator Log）を作成します。

```
# vxassist -g vvrdg make datavol_srl 4500M alloc=sdt
# vxassist -g vvrdg make archvol_srl 4500M alloc=sdu
```

メモ: プライマリ SRL と同じ名前の SRL を作成します。SRL をボリュームセットまたはボリュームセットのコンポーネントボリュームとすることはできません。

プライマリ RVG（Replicated Volume Group）にボリュームセットが含まれる場合は、追加するセカンダリのコンポーネントボリュームには必ずプライマリのコンポーネントボリュームと同じ名前、長さ、インデックスを設定するようにしてください。

Oracle を配備するためのプライマリとセカンダリのノード間の VVR の設定

プライマリとセカンダリのノード間に VVR を設定するには

1 プライマリのディスクグループ ID を取得します。

```
# vxprint -l  vvrdg
```

2 プライマリノードで /etc/vx/vras/.rdg ファイルを開きます。

3 /etc/vx/vras/.rdg ファイルでディスクグループ ID を追加します。

メモ: .rdg ファイルへの各グループの ID エントリが個別の行にあることを確認します。

---

4 セカンダリノードで /etc/vx/vras/.rdg ファイルを開きます。

5 /etc/vx/vras/.rdg ファイルでディスクグループ ID を追加します。

メモ: .rdg ファイルへの各グループの ID エントリが個別の行にあることを確認します。

---

6 プライマリノードから、ネットワークインターフェース上にランダムな IP アドレスがないかどうかを精査します。

7 セカンダリノードから、ネットワークインターフェース上にランダムな IP アドレスがないかどうかを精査します。

8 両方のノードからこれらの IP アドレスに ping を実行できるかどうかを検証します。

```
# ifconfig network_interface inet primary_IP netmask ¥
 subnet_mask up
# ifconfig network_interface inet secondary_IP netmask ¥
 subnet_mask up
```

9 レプリケートするすべてのボリュームに対して次のコマンドを使用して RDS（Replicated Data Set）のプライマリ RVG（Replicated Volume Group）を作成します。

```
# vradmin -g vvrdg createpri vvrdg_datavol_rvg datavol datavol_srl
# vradmin -g vvrdg createpri vvrdg_archvol_rvg archvol archvol_srl
```

10 プライマリ RVG と同じ名前のセカンダリ RVG を作成し、それらを RDS に追加します。

```
# vradmin -g vvrdg addsec vvrdg_datavol_rvg primary_IP ¥
 secondary_IP
# vradmin -g vvrdg addsec vvrdg_archvol_rvg primary_IP ¥
 secondary_IP
```

11 セカンダリノードにレプリケーションを開始します。

```
# vradmin -g vvrdg -a startrep vvrdg_datavol_rvg secondary_IP
# vradmin -g vvrdg -a startrep vvrdg_archvol_rvg secondary_IP
```

12 レプリケーションを停止します。

```
# vradmin -g vvrdg -f -s stoprep vvrdg_datavol_rvg secondary_IP
# vradmin -g vvrdg -f -s stoprep vvrdg_archvol_rvg secondary_IP
```

『Veritas InfoScale 7.4.2 レプリケーション管理者ガイド』を参照してください。

3

Storage Foundation for Databases (SFDB) ツールの設定

- 第7章 Storage Foundation for Databases リポジトリデータベースの設定および管理
- 第8章 Storage Foundation for Databases (SFDB) ツールの認証の設定

7

Storage Foundation for Databasesリポジトリデータベースの設定および管理

この章では以下の項目について説明しています。

- SFDB (Storage Foundation for Databases)リポジトリについて
- Storage Foundation for Databases (SFDB) ツールの必要条件
- SFDB（Storage Foundation for Database）ツールの可用性
- Storage Foundation for Databases (SFDB) リポジトリの設定
- Storage Foundation for Databases（SFDB）リポジトリのバックアップと復元
- ノード追加後の SFDB（Storage Foundation for Databases）リポジトリの更新
- ノード削除後の SFDB（Storage Foundation for Databases）リポジトリの更新
- SFDB (Storage Foundation for Databases) リポジトリの削除

SFDB (Storage Foundation for Databases)リポジトリについて

Storage Foundation for Databases（SFDB）リポジトリ、つまりリポジトリデータベースには、Storage Foundation for Databases で必要なメタデータ情報が格納されます。

メモ: Veritas InfoScale Enterprise 製品の初期インストールと設定以外で、リポジトリデータベースの操作が必要になるのはごくまれです。

Storage Foundation 製品のこのリリースでは、SFDB リポジトリはリレーショナルデータベースに格納され、SQLite3 によって管理されます。

Storage Foundation for Databases (SFDB) ツールの必要条件

製品の必要条件は、『Veritas InfoScale 7.4.2 リリースノート』に記載されています。

このハードウェア互換性リストにはサポートされているハードウェアの情報が記されており、定期的に更新されます。サポートされているハードウェアの最新情報については、次を参照してください。

https://www.veritas.com/content/support/en_US/doc/infoscale_hcl_73_731_74_unix

InfoScale 製品でサポートされている単一インスタンスの Oracle バージョンと Oracle RAC のバージョンについての最新情報については、次の場所にあるデータベースサポート表を参照してください。

https://www.veritas.com/bin/support/docRepoServlet?bookId=112632971-112632974-1&requestType=pdf

ハードウェアとソフトウェアの互換性を確認するには、現在の Oracle マニュアルを確認してください。

SFDB（Storage Foundation for Database）ツールの可用性

Oracle データベースの SFDB ツールは、次の製品に含まれています。

- 単一インスタンス Oracle を備えるホストシステムをサポートする Storage Foundation

  メモ: 必要な Veritas InfoScale Enterprise ライセンス。

- 自動フェールオーバーと単一インスタンス Oracle を備えるクラスタホストシステムをサポートする Veritas InfoScale Storage Foundation for Cluster File System HA
- 自動フェールオーバーと複数インスタンス Oracle を備えるクラスタホストシステムをサポートする Storage Foundation for Oracle RAC

このリリースでの SFDB ツールの機能変更および問題点について詳しくは、『Veritas InfoScale 7.4.2 リリースノート』で最新の詳細情報を参照してください。

Storage Foundation for Databases (SFDB) リポジトリの設定

Storage Foundation for Database (SFDB) リポジトリには SFDB ツールによって使われる情報が格納されます。リポジトリには、Storage Checkpoint の情報、表領域とデータファイルの情報、Database FlashSnap の情報、および SmartTier for Oracle のパラメータが含まれています。

SFDB リポジトリは、初めて `dbed_update or vxsfadm` を実行するときに自動で作成されます。

メモ: アップグレード時には、リポジトリを新しい形式に移行する必要があります。リリース 5.1 以降からアップグレードする場合は、このステップが必要です。

リポジトリがすでに存在する場合、そのコマンドはそのリポジトリを更新します。`dbed_update` を使用する場合:

- データベースが起動し、稼動している必要があります。
- ORACLE_SID と ORACLE_HOME の変数に -S と -H オプションが指定されている必要があります。

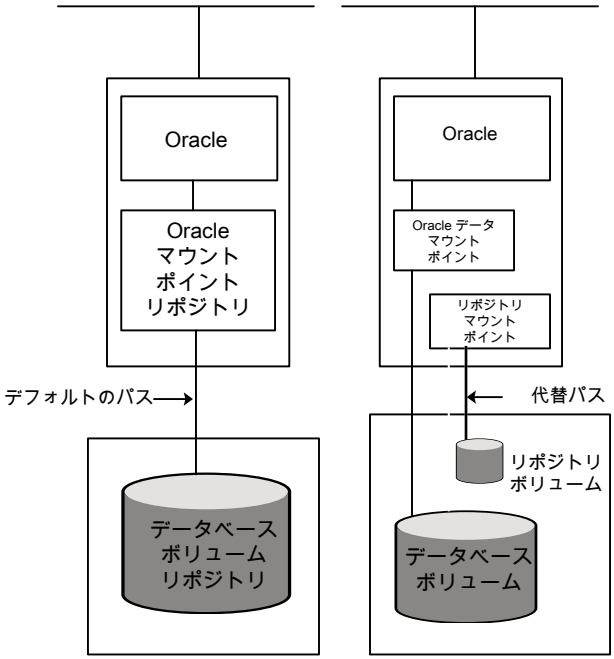
SFDB のリポジトリの場所

SFDB のリポジトリの場所は次のとおりです。

- デフォルトでは、SYSTEM 表領域の最初のデータファイルが格納されているディレクトリです。
- dbed_update で–R オプションを使って代替の場所を指定できます。

次の図では、リポジトリディレクトリは Oracle マウントポイントに存在します。

図 7-1 SFDB のリポジトリの場所



代替の場所には次の条件があります。

- Storage Foundation の場合、ローカルファイルシステム上にある必要があります
- Storage Foundation HA の場合、共有ストレージ上にあります。管理者は以下のことを行う必要があります。
  - フェールオーバーリソースを設定する
  - リポジトリと Oracle データベースがともに同じホストにフェールオーバーされるようにする
- Storage Foundation Cluster File System（HA）の場合、共有ストレージ上にあります。管理者は以下のことを行う必要があります。
  - Cluster File System またはフェールオーバーリソースを設定する
  - リポジトリと Oracle データベースがともに同じホストにフェールオーバーされるようにする
- Storage Foundation for Oracle RAC の場合、共有ストレージ上にあります。すべてのノードに CFS をマウントする必要があります。

メモ: 複数のデータベースで同じ代替パスを使うことはできません。

SFDB リポジトリの設定

dbed_update の使用法は次のとおりです。

```
dbed_update -S ORACLE_SID -H ORACLE_HOME
     [ -P ORACLE_PFILE ]
     [ -R REPOSITORY_PATH ] [ -I ]
```

次のオプションを使うことができます。

オプション	サンプル値	説明
-S	ORACLE_SID	情報を取得する Oracle データベースの名前を指定します。
-H	ORACLE_HOME	ORACLE_SID の ORACLE_HOME の設定です。
-P	ORACLE_PFILE	Oracle PFILE が ORACLE_HOME/dbs ディレクトリにない場合、Oracle PFILE の完全修飾パスを指定します。
-R	REPOSITORY_PATH	リポジトリのユーザー定義の場所を指定します。RAC または HA の状況では、共有ストレージに配置される必要があります。デフォルトの場所は、Oracle データベースの SYSTEM 表領域のマウントポイントです。このようにすると、クラスタノード間で簡単に共有されます。 REPOSITORY_PATH が「/」の場合、リポジトリは「/」に作成されません。SFDB ツールはリポジトリを「/」に作成することをサポートしていないため、リポジトリはデフォルトのパスに作成されます。
-I	N/A	Oracle データベースからの更新を行わずに、SFDB リポジトリの整合性チェックを実行します。成功した場合、文字列「ok」が stdout に出力されます。また、リポジトリファイルの絶対パスが表示されます。 データベースから更新する場合は、常に整合性チェックが自動的に実行されます。
-h	N/A	ヘルプページを表示します。

SFDB リポジトリを設定するには

◆ Oracle DBA のユーザーとして、次のコマンドを実行してデータベースの SFDB リポジトリを作成または更新します。

```
$ /opt/VRTS/bin/dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

Storage Foundation for Databases（SFDB）リポジトリのバックアップと復元

新しいFlashsnap 設定または Storage Checkpoint の作成などの重要な変更を行った後は、SFDB リポジトリのバックアップを作成する必要があります。

SFDB バックアップリポジトリの場合

- SFDB リポジトリと同じデフォルトの場所が使われます。
- 次のオプションを使って場所を指定できます。
  -d *directory*
- 異なるデータベースの SFDB リポジトリは、同じバックアップの場所を共有できません。
- バックアップ場所のディスク容量は自動的に管理されます。
- 各 SFDB リポジトリのバックアップファイルは、そのファイル名にタイムスタンプが含まれています。
- バックアップコマンドは、直前の 7 つのバックアップのみを保持します。

高可用性の環境の場合

- すべてのノードからバックアップ場所にアクセスできることを確認します。
- バックアップコマンドは任意のノードから実行できます。

`sfua_rept_util` の使用法

```
sfua_rept_util -S ORACLE_SID -o backup  [-d <directory>]
sfua_rept_util -S ORACLE_SID -o list  [-d <directory>]
sfua_rept_util -S ORACLE_SID -o restore  -f <backupfile> ¥
[-d <directory>]
```

表 7-1 `sfua_rept_util` **のオプション**

オプション	値	説明
-S	*ORACLE_SID*	リポジトリのバックアップが作成されるデータベースの SID を指定します。

オプション	値	説明
-o	backup	操作がバックアップの実行であることを指定します。
-o	list	操作がバックアップファイルの一覧表示であることを指定します。
-o	restore	操作が選択したバックアップファイルのリストアであることを指定します。
-d	*directory*	ファイルのバックアップまたはリストア用のディレクトリの場所を指定します。このオプションは必須ではありません。このオプションをバックアップに関して使う場合は、利用可能なバックアップとリストアの一覧表示用に使用する必要があります。
-f	*filename*	リストアされるリポジトリのバックアップのファイル名を指定します。このオプションは、バックアップファイルを復元する場合には必須です。

SFDB リポジトリのバックアップを作成するには

1. データベースがオンラインであり、バックアップ場所のディレクトリに DBA Oracle ユーザーが使用できる書き込み権限が付与されていることを確認します。
2. SFDB のリポジトリが最新の状態であることを確認します。

```
$ dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

3. Oracle DBA ユーザーとして、適切なオプションを指定して backup コマンドを実行します。

   次に例を示します。

```
$ /opt/VRTSdbed/bin/sfua_rept_util -S Prod -o backup
Backing up Repository for SID='Prod'
Target directory='/etc/vx/vxdba/logs/oragrid/backup'
Copying repository for SID=Prod to
'/etc/vx/vxdba/logs/oragrid/backup'
Backup Repository for SID='Prod' Completed
```

   oragrid は ORACLE_USER です。

SFDB リポジトリのバックアップを一覧表示するには

◆ Oracle DBA ユーザーとして list コマンドを実行します。

```
$ /opt/VRTSdbed/bin/sfua_rept_util -S Prod -o list
Listing Repository Backups for SID='Prod'
TIME                  SIZE              NAME
2012-06-12-19-43-55 14824 rept_Prod_2012-06-12-19-43-55.tgz
```

SFDB リポジトリを復元するには

◆ Oracle ユーザーとして、適切なオプションを指定して restore コマンドを実行します。

次に例を示します。

```
$ /opt/VRTSdbed/bin/sfua_rept_util -S Prod -o restore
-f rept_Prod_2012-06-12-19-43-55.tgz
Restoring Repository for SID='Prod'
Target directory='/etc/vx/vxdba/logs/oragrid/backup'
Restore Repository for SID='Prod' Completed
```

oragrid は ORACLE_USER です。

ノード追加後の SFDB（Storage Foundation for Databases）リポジトリの更新

クラスタにノードを追加した後に、SFDB リポジトリを更新して新しいノードのアクセスを有効にします。

ノードを追加した後に SFDB リポジトリを更新するには

1. クラスタ内のいずれかのノードから新しいノードに /var/vx/vxdba/rep_loc ファイルをコピーします。
2. 既存のクラスタノードに /var/vx/vxdba/auth/user-authorizations ファイルがある場合は、そのファイルを新しいノードにコピーします。

   既存のクラスタノードのどのノードにも /var/vx/vxdba/auth/user-authorizations ファイルがない場合は、必要な操作はありません。

   これで SFDB リポジトリへの新しいノードの追加が完了します。

ノード削除後の SFDB（Storage Foundation for Databases）リポジトリの更新

クラスタからノードを削除した場合は、SFDB リポジトリを更新する手順を実行する必要はありません。

製品を削除した後の SFDB リポジトリの削除について詳しくは、以下を参照してください。

p.72 の「SFDB (Storage Foundation for Databases) リポジトリの削除」を参照してください。

SFDB (Storage Foundation for Databases) リポジトリの削除

製品を削除した後に、SFDB リポジトリファイルとバックアップを削除できます。

SFDB リポジトリファイルを削除すると、SFDB ツールが無効になります。

SFDB リポジトリを削除するには

1 ホストで作成された SFDB リポジトリを特定します。

```
# cat /var/vx/vxdba/rep_loc

 {
    "sfae_rept_version" : 1,
    "oracle" : {
       "SFAEDB" : {
          "location" : "/data/sfaedb/.sfae",
          "old_location" : "",
          "alias" : [
             "sfaedb"
          ]
       }
    }
 }
```

2 `location` キーで特定されるディレクトリを削除します。

```
# rm -rf /data/sfaedb/.sfae
```

DB2 9.5 および 9.7:

```
# rm -rf /db2data/db2inst1/NODE0000/SQL00001/.sfae
```

DB2 10.1 および 10.5:

```
# rm -rf /db2data/db2inst1/NODE0000/SQL00001/MEMBER0000/.sfae
```

3 リポジトリの場所にあるファイルを削除します。

```
# rm -rf /var/vx/vxdba/rep_loc
```

これで SFDB リポジトリの削除が完了します。

8

Storage Foundation for Databases (SFDB) ツールの認証の設定

この章では以下の項目について説明しています。

- SFDB ツール認証のための vxdbd の設定
- SFDB ツールに対して認証を使っているクラスタへのノードの追加
- SFDB コマンドを実行できるようにするためのユーザーの許可

SFDB ツール認証のための vxdbd の設定

vxdbd を設定するには、root ユーザーとして次の手順を実行します

1 認証サービスを設定する sfae_auth_op コマンドを実行します。

```
# /opt/VRTS/bin/sfae_auth_op -o setup
Setting up AT
Starting SFAE AT broker
Creating SFAE private domain
Backing up AT configuration
Creating principal for vxdbd
```

2 vxdbd デーモンを停止します。

```
# /opt/VRTS/bin/sfae_config disable
vxdbd has been disabled and the daemon has been stopped.
```

3 /etc/vx/vxdbed/admin.properties 設定ファイルの AUTHENTICATION キーを yes に設定して、認証を有効にします。

/etc/vx/vxdbed/admin.properties が存在しない場合、cp /opt/VRTSdbed/bin/admin.properties.example /etc/vx/vxdbed/admin.properties を使用します。

4 vxdbd デーモンを起動します。

```
# /opt/VRTS/bin/sfae_config enable
vxdbd has been enabled and the daemon has been started.
It will start automatically on reboot.
```

vxdbd デーモンは認証を必要とするように設定されました。

SFDB ツールに対して認証を使っているクラスタへのノードの追加

SFDB ツール用の認証を使用してクラスタにノードを追加するには、root ユーザーとして次の手順を実行します

1 sfae_auth_op コマンドの -o export_broker_config オプションを使用して、すでに認証済みのクラスタ内のノードから、認証データをエクスポートします。

-f オプションを使用して、エクスポートされたデータを格納されるファイル名を指定します。

```
# /opt/VRTS/bin/sfae_auth_op ¥
-o export_broker_config -f exported-data
```

2 scp または rcp といった、利用できるコピー方法で、エクスポートされたファイルを新しいノードにコピーします。

3 sfae_auth_op コマンドの -o import_broker_config オプションを使用して、新しいノードに認証データをインポートします。

-f オプションを使用して、手順 2 でコピーしたファイルの名前を指定します。

```
# /opt/VRTS/bin/sfae_auth_op ¥
 -o import_broker_config -f exported-data
Setting up AT
Importing broker configuration
Starting SFAE AT broker
```

4 新しいノードの vxdbd デーモンを停止します。

```
# /opt/VRTS/bin/sfae_config disable
vxdbd has been disabled and the daemon has been stopped.
```

5 /etc/vx/vxdbed/admin.properties 設定ファイルの AUTHENTICATION キーを yes に設定して、認証を有効にします。

/etc/vx/vxdbed/admin.properties が存在しない場合、cp /opt/VRTSdbed/bin/admin.properties.example /etc/vx/vxdbed/admin.properties を使用します。

6 vxdbd デーモンを起動します。

```
# /opt/VRTS/bin/sfae_config enable
vxdbd has been enabled and the daemon has been started.
It will start automatically on reboot.
```

新しいノードは、クラスタと対話するように認証されて、SFDB コマンドを実行します。

SFDB コマンドを実行できるようにするためのユーザーの許可

セットアップまたはクラスタの各ノードでの SFDB コマンドの実行をユーザーに許可するには、root ユーザーとして次の手順を実行します。

◆ sfae_auth_op コマンドで -o auth_user オプションを使用して、ユーザー名を -u オプションの引数として指定します。

```
# /opt/VRTS/bin/sfae_auth_op ¥
-o auth_user -u oragrid
Creating principal oragrid@system1.example.com
```

上位ノードが同じクラスタの一部ではないオフホストのセットアップで、追加の -h オプションを使い、次の手順を実行します。

1 クラスタの一部である各プライマリノードで sfae_auth_op コマンドに -h *off-host hostname* を次の例のように追加します。

```
# /opt/VRTS/bin/sfae_auth_op -o auth_user -u oragrid -h myoffhost
Creating principal oragrid@dblxx64-2-v6.vxindia.veritas.com
```

2 オフホストノードで、sfae_auth_op コマンドに -h*primary node* を次の例のように追加します。

```
# /opt/VRTS/bin/sfae_auth_op -o auth_user -u oragrid -h
dblxx64-2-v6
Creating principal oragrid@myoffhost.vxindia.veritas.com
```

4

Oracle データベースのパフォーマンスの向上

- 第9章 データベースアクセラレータについて
- 第10章 Veritas Extension for Oracle Disk Manager によるデータベースパフォーマンスの向上
- 第11章 Veritas Cached Oracle Disk Manager によるデータベースパフォーマンスの向上
- 第12章 Quick I/O によるデータベースパフォーマンスの向上
- 第13章 Cached Quick I/O によるデータベースパフォーマンスの向上

9

データベースアクセラレータについて

この章では以下の項目について説明しています。

- Veritas InfoScale 製品コンポーネントのデータベースアクセラレータについて

Veritas InfoScale 製品コンポーネントのデータベースアクセラレータについて

どのような環境でも、相応のパフォーマンスを維持することや、パフォーマンスSLA（Service Level Agreements）を満たすことが主要な懸案事項です。Veritas InfoScale 製品コンポーネントは、様々な方法によってデータベース環境全体のパフォーマンスを向上させます。

表 9-1 Veritas InfoScale 製品コンポーネントのデータベースアクセラレータ

Veritas InfoScale データベースアクセラレータ	サポート対象のデータベース	使用例と注意事項
Oracle Disk Manager（ODM）	Oracle	■ ファイル I/O に対する最先端のカーネルサポートを含む改善されたアプリケーションプログラミングインターフェース（API）を利用して、Oracle のパフォーマンスを向上させ、システム帯域幅を管理します。 ■ Oracle Resilvering を使い、Veritas Volume Manager DRL（Dirty Region Logging）をオフにしてパフォーマンスを向上させるには、ODM を使います。 ■ ミッションクリティカルなアプリケーションの I/O 帯域幅をより多く利用できるようにして、データベースの一貫性を復元するために必要な時間を短縮するには、SmartSync Recovery Accelerator を使います。
Cached Oracle Disk Manager（Cached ODM）	Oracle	選択した I/O でキャッシュを使って ODM I/O のパフォーマンスを向上させるには、Cached ODM を使います。
Quick I/O（QIO）	Oracle DB2 Sybase	VxFS ファイルシステムで実行されるデータベースで RAW デバイスと同等のパフォーマンスを得るには、Quick I/O を使います。
Cached Quick I/O（Cached QIO）	Oracle DB2 Sybase	大容量のシステムメモリを活用して、頻繁にアクセスされるデータを選択的にバッファすることで、データベースのパフォーマンスをさらに向上させるには、Cached QIO を使います。
同時 I/O	DB2 Sybase	CIO（Concurrent I/O）は、DB2 環境および Sybase 環境向けに最適化されています。 ファイルサイズの拡張に関する制限なしで、VxFS ファイルシステム上で実行されるデータベースのパフォーマンスを向上させるには、Veritas InfoScale Concurrent I/O を使います。

これらのデータベースアクセラレータ技術によって、データベースのパフォーマンスは RAW ディスクパーティションと同等になりますが、さらにファイルシステムの管理性が得ら

れるという利点があります。Storage Foundation の DMP（Dynamic Multi-Pathing）機能により、サーバーからアレイまでのすべての使用可能なパスで I/O アクティビティの負荷を分散することで、パフォーマンスが最大限に引き上げられます。DMP は主要なハードウェア RAID ベンダーをすべてサポートしているので、サードパーティ製のマルチパスソフトウェアが不要となり、TCO を削減できます。

Veritas InfoScale データベースアクセラレータを使うと、データベースのパフォーマンスをより高精度で管理できるようになります。

ODM、Cached ODM、QIO、および Cached QIO for Oracle の使用について詳しくは、『Veritas InfoScale Oracle データベース用ストレージと可用性管理』を参照してください。

QIO、Cached QIO、および DB2 での同時 I/O の使用について詳しくは、『Veritas InfoScale DB2 データベース用ストレージと可用性管理』を参照してください。

ODM および Cached ODM for Oracle の使用について詳しくは、『Veritas InfoScale Oracle データベース用ストレージと可用性管理』を参照してください。

DB2 での同時 I/O の使用について詳しくは、『Veritas InfoScale DB2 データベース用ストレージと可用性管理』を参照してください。

10

Veritas Extension for Oracle Disk Manager によるデータベースパフォーマンスの向上

この章では以下の項目について説明しています。

- Veritas InfoScale 製品環境における Oracle Disk Manager について
- SFHA 環境の Veritas Extension for Oracle Disk Manager の設定
- SFHA 環境の Veritas Extension for Oracle Disk Manager の設定
- SFHA 環境で既存のデータベースストレージを Oracle Disk Manager 用として準備する方法
- SFHA 環境に Oracle Disk Manager を設定していることを確認
- SFHA 環境の Oracle Disk Manager 機能の無効化

Veritas InfoScale 製品環境における Oracle Disk Manager について

Veritas Extension for Oracle Disk Manager は、ファイル管理とディスクの I/O スループットを拡張します。Oracle Disk Manager の機能は、Veritas File System 環境で Oracle データベース用に最適化されます。Oracle Disk Manager を使うと、特殊な I/O 最適化により、I/O の多い作業負荷を持つデータベースの性能を改善できます。

Veritas Extension for Oracle Disk Manager は Oracle Resilvering をサポートします。Oracle Resilvering では、ストレージ層が Oracle データベースから、ミラー化データファイルのどの領域またはブロックをシステムクラッシュ後に再同期するのかという情報を受信します。Oracle Resilvering は、Oracle 自体がリカバリできるファイルに使用する Veritas Volume Manager Dirty Region Logging (DRL) のオーバーヘッドを回避して、パフォーマンスを向上させます。

Oracle Disk Manager では Oracle Managed Files の拡張サポートが用意されており、管理オーバーヘッドが減少します。Veritas Extension for Oracle Disk Manager には Quick I/O と同様の機能がありますが、ユーザーに対して透過的です。Veritas InfoScale Quick I/O とは異なり、Veritas Extension for Oracle Disk Manager を使って管理するファイルでは、特別なファイル命名規則が必要ありません。Oracle Disk Manager インターフェースは通常のデータベースファイルを使います。

Oracle Disk Manager では Oracle Managed Files の拡張サポートが用意されており、管理オーバーヘッドが減少します。Veritas Extension for Oracle Disk Manager はユーザーに対して透過的です。Oracle Disk Manager インターフェースは通常のデータベースファイルを使います。

メモ: Quick I/O は Linux ではサポートされません。

データベース管理者は、Oracle 製品で使うデータファイルのタイプを選択できます。従来、ファイルシステムに配置したファイルと RAW デバイスのどちらを選択するかは、管理性およびパフォーマンスに基づいて行われました。ただし、Oracle Parallel Server での使用を目的とするデータベースは例外です。この場合、多くのプラットフォームで RAW デバイスが必要です。パフォーマンスよりも管理性が重要な場合、一般に、ファイル形式としてファイルシステムのファイルが推奨されます。ただし、アプリケーションによっては、最初に実装されたときに I/O 必要条件が十分に満たせないような場合、I/O 必要条件が変更されることがあります。アプリケーションが I/O スループットに依存するようになると、ファイルシステムから RAW デバイスへのデータファイルの変換が必要な場合があります。

Oracle Disk Manager は、Oracle データベースと連携して高いパフォーマンスと管理性の両方を実現するように設計されています。Oracle Disk Manager は、VxFS ファイルシステム上や RAW ボリュームまたはパーティション上にあるデータベースストレージでの Oracle のファイル管理と I/O 呼び出しをサポートします。この機能は、動的にロードされる共有ライブラリとして提供され、この共有ライブラリはロードされるときに Oracle にバインドされます。Oracle Disk Manager ライブラリは、起動時にカーネルにロードされて、Oracle Disk Manager ドライバと連動します。

メモ: RAW デバイスは Linux ではサポートされません。

Oracle Disk Manager を使うメリットは次のとおりです。

- ファイルおよび RAW デバイスに対する本格的なカーネル非同期 I/O
- ファイルに対する本格的なカーネル非同期
- システムコールに対するオーバーヘッドの軽減
- VxFS ファイルシステムに隣接するようにファイルに領域を事前に割り当てることによるファイルシステムレイアウトの改善
- ファイルシステム上のファイルでの RAW デバイスと同じパフォーマンス
- ユーザーへの透過性

SFHA 環境の Oracle Disk Manager によるデータベースパフォーマンスの改善方法

Oracle Disk Manager では、次により、VxFS ファイルシステムに対するデータベースの I/O パフォーマンスを向上させます。

機能	説明
カーネルの非同期 I/O のサポート	非同期 I/O では、ブロッキングされていないシステムレベルの読み取りおよび書き込みが実行されます。これにより、システムは複数の I/O 要求を同時に実行できます。カーネル非同期 I/O は、コンテキストの切り替えを最小限にして作業を完了させるために、I/O がカーネル内のディスクデバイスドライバにキューイングされるため、ライブラリ非同期 I/O より優れています。
ダイレクト I/O のサポートと二重バッファリングの回避	read() および write() システムコールを使うファイルに対する I/O は、通常、データを 2 回コピーします。1 回目はユーザーとカーネル領域の間で、2 回目はカーネル領域とディスクの間で実行されます。対照的に、RAW デバイスの I/O は、ユーザー領域とディスク間で直接コピーされるため、1 回分のコピー操作が短縮されます。RAW デバイスの I/O と同様に、Oracle Disk Manager の I/O でも余分のコピーが回避されます。Oracle Disk Manager は、システムキャッシュをバイパスし、RAW デバイスと同じ効率でファイルにアクセスします。二重バッファリングの回避により、システム上のメモリオーバーヘッドを軽減できます。カーネルからユーザーのアドレス空間へのコピーが回避されるため、カーネルモードでのプロセッサの使用率が大幅に軽減され、解放されたプロセッササイクルによって今までよりも多くのアプリケーションコードを実行できます。

機能	説明
データベースファイルに対するカーネル書き込みロックの回避	write() システムコールを使ってデータベース I/O を実行すると、システムコールごとに、ファイルに対するカーネルの書き込みロックの取得および解放が行われます。このロックは書き込み中に読み込みをブロックし、同じファイルへの同時書き込み操作を防ぎます。通常、データベースシステムには、ファイルに対する同時アクセスを管理する独自のロック機能が実装されているため、ファイルごとの書き込みロックは不要な I/O 操作です。Oracle Disk Manager は、ファイルシステムによるロックをバイパスし、データベースサーバーがデータアクセスを制御できるようにします。
1 回のシステムコールでの複数の同時 I/O のサポート	非同期 I/O を実行すると、Oracle プロセスは I/O の完了を待つ間は他に実行できる処理が存在しないため、完了を待ちながら追加 I/O 要求を発行したり、特定の I/O 要求を待機したりする場合があります。また、Oracle プロセスは別のファイルに対して要求を発行する場合もあります。このすべての動作は、Oracle で Oracle Disk Manager I/O インターフェースが使われていれば、1 回のシステムコールで実現できます。このインターフェースによって、同じ作業を完了するために実行されるシステムコール数が削減されるため、プロセス内のユーザー空間とカーネル空間のコンテキスト切り替え回数が減ります。 **メモ:** `FILESYSTEMIO_OPTIONS` 初期化パラメータを使って、ファイルシステムのファイルの非同期 I/O、ダイレクト I/O、同時 I/O の有効と無効を切り替えることができます。このパラメータは JFS ファイルと JFS2 ファイルのみに適用されます。このパラメータは VxFS ファイル、ODM ファイル、QIO ファイルには適用されません。詳しくは、Oracle が提供する Oracle Disk Manager のマニュアルを参照してください。
複数の Oracle インスタンスによるファイルの二重オープンの回避	Oracle Disk Manager では、「ファイル識別子」が用意されており、ファイルを 1 回のみ開くことが許されます。これはファイルの「識別」と呼ばれます。同じファイル識別子を、Oracle インスタンス内の他のプロセスで使うことができます。ファイルの状態は、カーネル内の Oracle Disk Manager ドライバによって保持されます。ファイルのオープンコール数が減少することで、プロセスの初期化時と終了時の処理オーバーヘッドが減少し、また、カーネル内で必要なファイル状態の構造数が減少します。

機能	説明
連続するデータファイルの割り当て	Oracle Disk Manager は、ソートクエリーや並列クエリーなどの一時表領域を使うクエリーのパフォーマンスを向上させることができます。Oracle Disk Manager を使わない場合、Oracle は一時表領域用のデータファイルを初期化しません。したがって、データファイルはスパースファイルになり、通常は断片化されます。スパースファイルまたは断片化されたファイルでは、クエリーパフォーマンスが低下します。Oracle Disk Manager を使うと、データファイルは一時表領域用に初期化されて連続的に割り当てられるため、スパースファイルにはなりません。

SFHA 環境の SmartSync について

SmartSync Recovery Accelerator は、変更されたデータのみを再同期化することで、ミラーボリュームの可用性を向上させます。SmartSync によって、ミッションクリティカルなアプリケーションの I/O 帯域幅をより多く利用できるようにすることで、データベースの一貫性を復元するために必要な時間が短縮されます。

Oracle が RAW ボリュームインターフェースをサポートしていないため、SmartSync は RAW ボリューム上に構成された Oracle データベースをサポートしていません。

ODM の機能が拡張され、ファイルレベルで SmartSync を使用できるようになりました。この機能は ODM Resilvering と呼ばれます。設定は必要ありません。デフォルトで利用可能です。

Oracle Disk Manager と Oracle Managed Files の連携

次の例は、Oracle Disk Manager と Oracle Managed Files (OMF) との関係を示しています。例では `init.ora` の内容とデータベースインスタンスを起動するコマンドを示します。Oracle の UNDO の管理を簡略化するために `init.ora` パラメータ `UNDO_MANAGEMENT` は `AUTO` に設定されます。これは、システム管理 UNDO と呼ばれます。

メモ: OMF データベースを構築する前に `init.ora` の適切なデフォルト値が必要です。この値により `CREATE DATABASE` 文の実行後に `SYSTEM` 表領域、オンライン REDO ログ、制御ファイルの場所を制御できます。

```
$ cat initPROD.ora
UNDO_MANAGEMENT = AUTO
DB_CREATE_FILE_DEST = '/PROD'
DB_CREATE_ONLINE_LOG_DEST_1 = '/PROD'
db_block_size = 4096
```

```
db_name = PROD
$ sqlplus /nolog
SQL> connect / as sysdba
SQL> startup nomount pfile= initPROD.ora
```

Oracle インスタンスが起動します。

```
Total System Global Area 93094616 bytes
Fixed Size 279256 bytes
Variable Size 41943040 bytes
Database Buffers 50331648 bytes
Redo Buffers 540672 bytes
```

EMP_TABLE 表領域に関連付けられたファイルを EMP_INDEX 表領域とは別のディレクトリに配置するレイアウトを実装するには、ALTER SYSTEM 文を使います。この例では、OMF によるファイル名、storage 句とパスの取り扱い方法を示します。このレイアウトにより、表領域をデータファイルの集合ではなくファイルシステム内のオブジェクトと見なすことができます。OMF では、Oracle Disk Manager のファイルサイズ変更機能が使われるため、表領域ファイルは、デフォルトサイズ 100 MB で最初に作成され、必要に応じて拡張されます。拡張を制限するには MAXSIZE 属性を使います。

次の例に、OMF データベースを作成するコマンドと、それぞれの場所に EMP_TABLE および EMP_INDEX 表領域を作成するコマンドを示します。

メモ: OMF が機能する上でディレクトリが存在する必要があるため SQL*Plus HOST コマンドを使ってディレクトリを作成しています。

```
SQL> create database PROD;
```

データベースが作成されます。

```
SQL> HOST mkdir /PROD/EMP_TABLE;
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/PROD/EMP_TABLE';
```

システムが変更されました。

```
SQL> create tablespace EMP_TABLE DATAFILE AUTOEXTEND ON MAXSIZE ¥
500M;
```

表領域が作成されました。

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST = '/PROD/EMP_INDEX';
```

システムが変更されました。

```
SQL> create tablespace EMP_INDEX DATAFILE AUTOEXTEND ON MAXSIZE ¥
100M;
```

表領域が作成されました。

ls コマンドを使って、新しく作成されたデータベースを表示します。

```
$ ls -lFR
total 638062
drwxr-xr-x 2 oracle11g dba 96 May  3 15:43 EMP_INDEX/
drwxr-xr-x 2 oracle11g dba 96 May  3 15:43 EMP_TABLE/
-rw-r--r-- 1 oracle11g dba 104858112 May 3 17:28 ora_1_BEhYgc0m.log

-rw-r--r-- 1 oracle11g dba 104858112 May 3 17:27 ora_2_BEhYu4NA.log
-rw-r--r-- 1 oracle11g dba 806912 May 3 15:43 ora_BEahlfUX.ctl
-rw-r--r-- 1 oracle11g dba 10489856 May 3 15:43
ora_sys_undo_BEajPSVq.dbf
-rw-r--r-- 1 oracle11g dba 104861696 May 3 15:4
ora_system_BEaiFE8v.dbf
-rw-r--r-- 1 oracle11g dba 186 May 3 15:03 PROD.ora

./EMP_INDEX:
total 204808
-rw-r--r-- 1 oracle11g dba 104861696 May 3 15:43
ora_emp_inde_BEakGfun.dbf

./EMP_TABLE:
total 204808
-rw-r--r-- 1 oracle11g dba 104861696 May 3 15:43
ora_emp_tabl_BEak1LqK.dbf
```

SFHA 環境の Veritas Extension for Oracle Disk Manager の設定

Veritas Extension for Oracle Disk Manager を設定するには、次の条件を満たす必要があります。

前提条件	■ Veritas InfoScale Storage Foundation Enterprise 製品または Standard 製品がシステムにインストールされている必要があります。 ■ サポートされている Oracle データベースのバージョンがシステムにインストールされている必要があります。 ■ Cached Quick I/O が使用可能であり、データファイルに対して Cached Quick I/O が有効になっている場合は、Oracle Disk Manager を有効にしないでください。
使用に関する注意事項	Quick I/O 機能が使用可能な場合、Oracle Disk Manager は Quick I/O ドライバを使って非同期 I/O を実行します。デフォルトである Quick I/O のマウントオプションをオフにしないでください

SFHA 環境の Veritas Extension for Oracle Disk Manager の設定

`ORACLE_HOME` が共有ファイルシステムにある場合は、任意のノードから次の手順を実行します。それ以外の場合は、各ノードで次の手順を実行します。*`ORACLE_HOME`* は Oracle データベースバイナリがインストールされている場所です。

Veritas Extension for Oracle Disk Manager を設定するには

1. `oracle` としてログインします。
2. Oracle データベースが動作している場合は、Oracle データベースをシャットダウンします。
3. Oracle データベースのシャットダウン後に ODM モジュールを無効にします。

   AIX の場合:

   ```
   # /etc/rc.d/rc2.d/S99odm stop
   ```

   サポート対象の Linux 配布の場合:

   ```
   # systemctl stop vxodm
   ```

   Solaris の場合:

   ```
   # /usr/sbin/svcadm disable vxodm
   ```

4 次のライブラリが存在することを確認します。

AIX の場合:

```
/opt/VRTSodm/lib/libodm64.so
```

Linux の場合:

```
/opt/VRTSodm/lib64/libodm.so
```

Solaris SPARC の場合:

```
/opt/VRTSodm/lib/sparcv9/libodm.so
```

5 ORACLE_HOME にある Oracle の ODM ライブラリに Veritas Extension for Oracle Disk Manager ライブラリをリンクします。

サポート対象の Oracle のバージョン

- $ORACLE_HOME/lib ディレクトリに移動します。

```
# cd $ORACLE_HOME/lib
```

- libodm18.so のバックアップを作成します。

```
# mv libodm<version>.so
libodm<version>.so.oracle-`date '+%m_%d_%y-%H_%M_%S'`
```

- libodm<*version*>.so を Veritas ODM ライブラリとリンクします。

```
$ cd $ORACLE_HOME/rdbms/lib
$ mkdir -p odm
$ cd $ORACLE_HOME/rdbms/lib/odm
```

AIX の場合:

ln -s /opt/VRTSodm/lib64/libodm.so libodm<*version*>.so

Linux の場合:

ln -s /opt/VRTSodm/lib/libodm64.so libodm<*version*>.so

Solaris SPARC の場合:

ln -s /usr/lib/sparcv9/libodm.so libodm<*version*>.so

6 ODM モジュールをロードします。

AIX の場合:

```
# /etc/rc.d/rc2.d/S99odm start
```

サポート対象の Linux 配布の場合:

```
# systemctl start vxodm
```

Solaris の場合:

```
# /usr/sbin/svcadm enable vxodm
```

7 Oracle データベースを起動します。

8 Oracle データベースが Veritas Extension for ODM とともに開始することを確認するために、警告ログには次のテキストが含まれます。

```
Oracle instance running with ODM:
 Veritas 7.4.2.0000 ODM Library, Version 2.0
```

SFHA 環境で既存のデータベースストレージを Oracle Disk Manager 用として準備する方法

VxFS ファイルシステム内の Quick I/O 以外のファイルは、Oracle Disk Manager と連携するために、一切の変更をする必要はありません。ファイルは、デフォルトで Oracle Disk Manager I/O として検出され識別されます。Oracle Disk Manager データファイルを最大限に活用するには、ファイルを断片化しないでください。

VxFS ファイルシステム内で使用している Quick I/O ファイルを Oracle Disk Manager に移動する場合、Quick I/O ファイルを通常のファイルに変換してください。

メモ: Quick I/O は Linux ではサポートされません。

VxFS ファイルシステム内のファイルは、Oracle Disk Manager と連携するために一切の変更を必要としません。ファイルは、デフォルトで Oracle Disk Manager I/O として検出され識別されます。Oracle Disk Manager データファイルを最大限に活用するには、ファイルを断片化しないでください。

SFHA 環境に Oracle Disk Manager を設定していることを確認

Oracle Disk Manager が設定されていることを検証する前に、次の条件を満たしていることを確認します。

AIX の前提条件

- /opt/VRTSodm/lib/libodm64.so が存在する必要があります。
- $ORACLE_HOME/rdbms/lib/odm/libodm*version*.so が /opt/VRTSodm/lib/libodm64.soにリンクされています。
- VRTSodm パッケージ がインストールされている必要があります。

Linux の前提条件

- /opt/VRTSodm/lib64/libodm.so が存在する必要があります。
- $ORACLE_HOME/rdbms/lib/odm/libodm*version*.so が /opt/VRTSodm/lib64/libodm.soにリンクされています。
- VRTSodm パッケージ がインストールされている必要があります。

Solaris の前提条件

- /opt/VRTSodm/lib/libodm.so が存在する必要があります。
- $ORACLE_HOME/rdbms/lib/odm/libodm*version*.soは /opt/VRTSodm/lib/sparcv9/libodm.soにリンクされています。
- VRTSodm パッケージがインストールされている必要があります。

Oracle Disk Manager が設定されていることを検証するには

1 ODM 機能がライセンスに含まれていることを検証します。

```
# /opt/VRTS/bin/vxlicrep | grep ODM

QLOGODM                     = Enabled
   ODM                                  = Enabled
```

出力は ODM が有効になっていることを検証します。

メモ: ODM 機能を含んでいるライセンスキーが期限切れになっていないことを検証します。ライセンスキーが期限切れになっている場合、ODM 機能は使えません。

---

2 `VRTSodm` パッケージ をインストールしていることを確認します。

Solaris の場合:

```
# pkginfo VRTSodm
```

AIX の場合:

```
# lslpp -L VRTSodm
```

Linux の場合:

```
# rpm -qa | grep VRTSodm
```

または

```
# rpm -qi VRTSodm
```

3 Veritas ODM ライブラリ、`libodm64.so` または `libodm.so` が存在することを確認します。たとえば:

AIX の場合は、`# ls -lL /opt/VRTSodm/lib/libodm64.so`を使用します。

Linux の場合は、`# ls -lL /opt/VRTSodm/lib/libodm64.so`を使用します。

Solarisの場合は、`# ls -lL /usr/lib/sparcv9/libodm.so`を使用します。

Oracle Disk Manager が実行されていることを検証するには

1 Oracle データベースを起動します。

2 インスタンスが Oracle Disk Manager 機能を使っていることを確認します。

```
# cat /dev/odm/stats
# echo $?
0
```

3 Oracle Disk Manager が読み込まれていることを検証します。

Solaris の場合:

```
# modinfo | grep ODM | grep VRTS
293 7b6d2000 18078  30   1   odm (VRTS ODM 7.4.1.0,REV=internal-b)
```

Linux の場合:

```
# lsmod | grep odm
vxodm   164480  1
fdd 78976 1 vxodm
```

AIX の場合:

genkld または genkex コマンドを使うことができます。

```
# genkld | grep odm
or
# genkex | grep odm


        d05a3100    22182 /usr/lib/libodm.a[shr.o]
 90000001c0ec000    1675 /ora_base/db_home/lib/libodm12.so
 900000000bbf000    1059e /opt/VRTSodm/lib/libodm64.so
 90000000043d300    24167 /usr/lib/libodm.a[shr_64.o]
bash-3.2# genkex | grep odm
f1000000c037a000    27000 /usr/lib/drivers/vxodm.ext_71
```

4 警告ログで、Oracle インスタンスが実行されていることを検証します。ログに次のような出力が含まれている必要があります。

```
Oracle instance running with ODM:
Veritas 7.4.1.0000 ODM Library, Version 2.0
```

SFHA 環境の Oracle Disk Manager 機能の無効化

Oracle Disk Manager 機能は通常のファイルを使うので、この機能を無効にすることにより、すぐに通常の VxFS ファイルとしてファイルにアクセスできます。

Oracle Disk Manager 機能を無効にする手順は、32 ビット版および 64 ビット版の Oracle ともに同じです。

メモ: Quick I/O は Linux ではサポートされません。

メモ: Quick I/O を使う VxFS に変換するには、次の手順に従って Oracle Disk Manager を無効にし、ファイルを Quick I/O ファイルに変換します。

p.124 の「Oracle ファイルの Quick I/O ファイルへの変換」を参照してください。

Oracle Disk Manager 機能を無効にする前に、ファイルのバックアップを作成しておくと安全です。

Oracle インスタンス内の Oracle Disk Manager 機能を無効にするには

1. データベースインスタンスを停止します。
2. AIX では `rm` コマンドと `ln` コマンドを使って、Oracle Disk Manager ライブラリへのリンクを削除します。

```
$ rm $ORACLE_HOME/rdbms/lib/odm/libodmversion.so
$ ln -s ${ORACLE_HOME}/lib/libodmdversion.so ¥
${ORACLE_HOME}/lib/libodmversion.so
```

3. Linux では `rm` コマンドと `ln` コマンドを使って、Oracle Disk Manager ライブラリへのリンクを削除します。

```
$$ORACLE_HOME/rdbms/lib/odm/libodmversion.so
$ ln -s ${ORACLE_HOME}/lib/libodmdversion.so ¥
${ORACLE_HOME}/lib/libodmversion.so
```

4. Solaris では `rm` コマンドと `ln` コマンドを使って、Oracle Disk Manager ライブラリへのリンクを削除します。

```
$ rm $ORACLE_HOME/rdbms/lib/odm/libodmversion.so
$ ln -s ${ORACLE_HOME}/lib/libodmdversion.so ¥
${ORACLE_HOME}/lib/libodmv.so
```

5. データベースインスタンスを再起動します。

11

Veritas Cached Oracle Disk Manager によるデータベースパフォーマンスの向上

この章では以下の項目について説明しています。

- SFHA 環境の Cached ODM について
- SFHA 環境の Cached ODM の設定
- SFHA 環境の Cached ODM Advisor による Cached ODM 設定の管理

SFHA 環境の Cached ODM について

ODM I/O は通常ファイルシステムキャッシュをバイパスし、ディスクから直接読み込み、ディスクに直接書き込みます。Cached ODM によって一部の I/O はキャッシュを使用して先読みを行うことができ、特定の作業負荷に対して ODM I/O パフォーマンスを改善できます。Cached ODM は、Oracle からの I/O ごとのヒントを基に条件形式のキャッシュを実行できます。ヒントは Oracle がデータに対して何を行うかを示します。ODM はこれらのヒントを使用してキャッシュを実行し、一部の読み取りで先読みを行いますが、ODM は同じファイルであってもその他の読み込みではキャッシュを使用しません。

SFHA 環境の Cached ODM の使用に関する注意事項

Cached ODM は、読み取りが集中するような作業負荷に対して最も役立ちます。書き込みが集中するような作業負荷やメモリが不足している環境では、Cached ODM は推奨されません。

ODM に優る Cached ODM の利点は次のとおりです。

- ODM はダイレクト I/O を実行します。
  - Oracle では、より大きいシステムグローバル領域（SGA）を使うことができます。
  - 先読みはサポートされていません。
- Oracle の読み取りが集中するような作業負荷によって、ODM の使用時にパフォーマンスが低下する可能性があります。
- ホストに複数のデータベースが存在する場合は、Oracle SGA が不足することがあります。
  - ホストには複数のデータベースを含めることができます。（ページキャッシュによって複数のデータベースに対応できます。）
  - パラレルクエリープロセスの多くは SGA を使いません。

p.98 の「SFHA 環境のファイルシステムに対する Cached ODM の有効化」を参照してください。

SFHA 環境の Cached ODM でサポートされている設定

Cached ODM は、次の環境でサポートされています。

- Storage Foundation（HA）
- Storage Foundation Cluster File System（HA）
- ローカルマウントおよびクラスタマウンド

現在、Storage Foundation for Oracle RAC はサポートされていません。

Cached ODM はキャッシュを有効にしなかったファイルおよびファイルシステムのパフォーマンスに影響しません。

SFHA 環境の Cached ODM Advisor について

Cached ODM Advisor はデータベース管理者用のツールで、次の機能を提供します。

- Cached ODM Advisor により、root 権限を必要とせずに、データベースファイルで Cached ODM を管理、設定できます。
- Cached ODM Advisor によって適用された設定は、すべてのクラスタノードにわたって、その一貫性と共通性が自動的に維持されます。

メモ: Cached ODM Advisor は、ODM cachemap へのインターフェースを提供していません。

SFHA 環境の Cached ODM の設定

Cached ODM を設定するには、最初にファイルシステムに対して Cached ODM を有効にする必要があります。Cached ODM を有効にしたら、Cached ODM を次の 2 つの方法で設定できます。

- 1 つ目の設定方法: ファイルごとにすべての I/O のキャッシュをオンまたはオフにします。
- 2 つ目の設定方法: ODM の cachemap を調整します。cachemap はファイルのタイプと I/O のタイプの組み合わせをキャッシュアドバイザリへとマップします。

p.99 の「SFHA 環境の個々のファイルの Cached ODM 設定の調整」を参照してください。

p.100 の「SFHA 環境の cachemap を使った Cached ODM 設定のチューニング」を参照してください。

SFHA 環境のファイルシステムに対する Cached ODM の有効化

ファイルシステムに対する Cached ODM は最初は無効になっています。ファイルシステムのマウント後に、`vxtunefs` コマンドの `odm_cache_enable` オプションを設定して、ファイルシステムに対する Cached ODM を有効にすることができます。

`vxtunefs`(1M)のマニュアルページを参照してください。

メモ: この `vxtunefs` コマンドを使うと、このファイルシステム上のすべての ODM ファイルのキャッシュが有効になります。

ファイルシステムの Cached ODM を有効にするには

1. VxFS ファイルシステム `/database01` で Cached ODM を有効にします。

```
# vxtunefs -o odm_cache_enable=1 /database01
```

2. `/etc/vx/tunefstab` ファイルにファイルシステムエントリを追加すると、以後のマウントでもこの設定を有効にできます。

```
/dev/vx/dsk/datadg/database01 odm_cache_enable=1
```

   `tunefstab`(4)マニュアルページを参照してください。

3. Cluster File System では、すべてのノードで /etc/vx/tunefstab を変更する必要があります。

SFHA 環境の個々のファイルの Cached ODM 設定の調整

odmadm setcachefile コマンドを使用すると、特定のファイルの cachemap を無視して、ODM がファイルへの I/O をすべてキャッシュに保存する、またはいずれもキャッシュに保存しないようにできます。キャッシュ状態は、ON、OFF、DEF（デフォルト）のいずれかにできます。キャッシュ状態 DEF は条件キャッシュで、I/O ごとに、ODM は cachemap を調べ、指定されたファイルのタイプと I/O のタイプの組み合わせをキャッシュに保存する必要があるかどうかを判断します。キャッシュ状態 ON は、指定されたファイルを常にキャッシュに保存し、キャッシュ状態 OFF は指定されたファイルをキャッシュに保存しません。

odmadm（1M）のマニュアルページを参照してください。

メモ: ファイルシステムに対する Cached ODM を有効にしている場合にかぎり、キャッシュアドバイザリが機能します。odm_cache_enable フラグが 0 に設定されている場合は、特定のファイルのキャッシュアドバイザリを有効（ON）に設定しても、そのファイルシステムのすべてのファイルに対する Cached ODM が無効（OFF）になります。

特定のファイルで無条件キャッシュを有効にする方法

◆ /mnt1/file1 ファイルで無条件キャッシュを有効にします。

```
# /opt/VRTS/bin/odmadm setcachefile /mnt1/file1=on
```

このコマンドにより、ODM は file1 からのすべての読み取りをキャッシュに保存します。

特定のファイルでキャッシュを無効にする方法

◆ /mnt1/file1 ファイルでキャッシュを無効にします。

```
# /opt/VRTS/bin/odmadm setcachefile /mnt1/file1=off
```

このコマンドにより、ODM は file1 からの読み取りをキャッシュに保存しません。

ファイルの現在のキャッシュアドバイザリの設定を確認する方法

◆ /mnt1/file1 および /mnt2/file2 ファイルの現在のキャッシュアドバイザリ設定を確認します。

```
# /opt/VRTS/bin/odmadm getcachefile /mnt1/file1 /mnt2/file2
/mnt1/file1,ON
/mnt2/file2,OFF
```

すべてのファイルをデフォルトのキャッシュアドバイザリにリセットする方法

◆ すべてのファイルをデフォルトのキャッシュアドバイザリにリセットします。

```
# /opt/VRTS/bin/odmadm resetcachefiles mountpoint
```

SFHA 環境の cachemap を使った Cached ODM 設定のチューニング

odmadm setcachemap コマンドを使って、cachemap を設定できます。cachemap は、ファイルタイプと I/O タイプの組み合わせをキャッシュアドバイザリにマップします。ODM は、デフォルトの条件付きキャッシュ設定を持つすべてのファイルに対して cachemap を使います。これらのファイルは、odmadm setcachefile コマンドによってキャッシュ処理がオンまたはオフにされていないファイルです。

odmadm(1M)のマニュアルページを参照してください。

デフォルトでは、cachemap は空です。ただし、odmadm setcachemap コマンドを使ってキャッシュアドバイザリを追加できます。

キャッシュアドバイザリを cachemap に追加するには

1. キャッシュアドバイザリを cachemap に追加するには、次のコマンドを実行します。

   ```
   # /opt/VRTS/bin/odmadm setcachemap ¥
    data/data_read_seq=cache,readahead
   ```

   このコマンド例では、ODM は I/O にキャッシュ処理と readahead を使って、data_read_seq I/O タイプを持つオンラインログファイル(data)をオンラインにします。odmadm getcachemap コマンドの出力で、有効なファイルタイプと I/O タイプの値を確認できます。

   odmadm(1M)のマニュアルページを参照してください。

2. cachemap はノードごとにローカルです。クラスタの各ノードでキャッシュ判定を同一にするには、クラスタ内の各ノードで同じ /opt/VRTS/bin/odmadm setcachemap コマンドを実行して、cachemap の設定の一貫性を保ちます。

SFHA 環境のマウント全体を通したキャッシュ設定の永続化

デフォルトでは、Cached ODM 設定はマウント全体をとおして永続的ではありません。それらの設定を odmadm 設定ファイルに追加することで、設定を永続化できます。

マウント全体をとおしてキャッシュ設定を永続化にするには

1 cachemap の設定を /etc/vx/odmadm に追加できます。

```
setcachemap data/read_data_header=cache
setcachemap all/datapump=cache,readahead
```

2 cachemap はノードごとにローカルです。クラスタ内で cachemap の一貫性を保つには、/etc/vx/odmadm の内容がすべてのノードで一致している必要があります。

3 ファイルシステムの lost+found/odmadm ファイルに、ファイルごとの設定を手動で追加できます。

たとえば、ファイルシステムがマウントされるたびに oradata/file1 のキャッシュを無効にするには、ファイルシステムの lost+found/odmadm ファイルに次のとおりに入力します。

```
setcachefile oradata/file1=off
```

lost+found/odmadm ファイルのファイルごとの設定は、Cached ODM Manager を使って管理することもできます。Cached ODM Manager を使う場合は、このファイルを手動で編集しないでください。

p.109 の「SFHA 環境の Cached ODM Advisor を使用したデータファイルの Cached ODM の有効化/無効化」を参照してください。

SFHA 環境の Cached ODM Advisor による Cached ODM 設定の管理

Cached ODM Advisor は、データベース管理者が行う Cached ODM 設定の管理タスクを簡素化します。

- Cached ODM Advisor により、root 権限を必要とせずに、データベースファイルで Cached ODM を管理、設定できます。
- Cached ODM Advisor によって適用された設定は、すべてのクラスタノードにわたって、その一貫性と共通性が自動的に維持されます。
- Cached ODM Advisor は ODM の cachemap に対するインターフェースとはなりませんが、Cached ODM 設定を ON/OFF に設定することができます（DEF には設定できません）。
- ODM Advisor により、Cached ODM を有効にする場合に適したデータファイルを判断するのに役立つ、さまざまなレポートを生成できます。

Cached ODM Advisor のコマンド `dbed_codm_adm` は、DBA ユーザーが実行してください。

メモ: クローンデータベースで Cached ODM Advisor を使用する場合、クローンデータベースは、プライマリデータベースとして有効になっていると Cached ODM 設定を有効にしません。クローンデータベースで `odm_cache_enable = 1` と設定することにより、Cached ODM を手動で有効にする必要があります。

Cached ODM Advisor のコマンド `dbed_codm_adm` の構文を以下に示し、そのオプションの一覧を 表 11-1 に示します。

```
Cached ODM Administration
dbed_codm_adm
Usage:
-S ORACLE_SID -H ORACLE_HOME -o display [ -n num -c col -t tbs ]
-S ORACLE_SID -H ORACLE_HOME -o display [ -n num ] [ -t tbs ] [-p] [-a]
                            [ -d start_date end_date
                              -m start_time end_time ]
                            [ -b blacklist_file ]
                            [ datafile ]
-S ORACLE_SID -H ORACLE_HOME -o display [ -n num ] [ -t tbs ] [-p] [-a]
                            [ -u snapshot_range [ -u snapshot_range ] ]
                            [ -b blacklist_file ]
                            [ datafile ]
-S ORACLE_SID -H ORACLE_HOME -o display [ -n num ] [ -u num_snapshots]
                            [ -t tbs ]
                            [ -b blacklist_file ]
                            [ datafile ]
-S ORACLE_SID -H ORACLE_HOME -o [ on | off | odmstats ] datafile | -f filelist
-S ORACLE_SID -H ORACLE_HOME -o add-to-blacklist { datafile | -f filelist }
-S ORACLE_SID -H ORACLE_HOME -o remove-from-blacklist { datafile | -f filelist }
-S ORACLE_SID -H ORACLE_HOME -o print-blacklist
-S ORACLE_SID -H ORACLE_HOME -o filestate [ datafile | -f filelist ]
-o iostats
```

表 11-1 Cached ODM Advisor のコマンドのオプション

オプション	用途
-S ORACLE_SID	ORACLE_SID を指定します。ORACLE_SID は、使用する Oracle データベースの SID です。
-H ORACLE_HOME	ORACLE_HOMEを指定します。ORACLE_HOME は、使用する Oracle インスタンスの HOME です。
-o display	データファイルでの読み取りアクティビティのレポートを生成します。

オプション	用途
-o filestate	データファイルで Cached ODM がオンになっているかどうかを表示します。
-o on	Cached ODM を有効にします。
-o off	Cached ODM を無効にします。
-o odmstats	データファイルの Cached ODM I/O 統計情報を表示します。
-o iostats	ファイルタイプと I/O タイプの組み合わせに関して累積の I/O 統計情報を表示します。
-c column_index	基本レポートを保存するカラムを指定します。これはオプションのフィールドです。デフォルトの I/O 統計情報は、物理的な読み取り回数（PHYRDS）でソートされます。
-n num	レポートに含まれるデフォルトのファイル数を変更します。このオプションを `-o display` とともに使うと、Cached ODM を有効にする上位 num 個の候補ファイルが表示されます。
-f filelist	操作するデータファイルのリストを含むファイル名を指定します。このリストには各行に 1 つのファイル名が含まれています。リストを指定しないと、`-o on` オプションまたは `-o off` オプションの場合を除いて、すべてのデータファイルが処理されます。
-t tbs	レポートを生成する表領域の名前を指定します。
filename	処理する単一ファイルの名前を指定します。
-d start_date end_date	生成するレポートの日付範囲を指定します。日付には範囲指定に使用した日付も含まれ、yyyy/mm/dd 形式で指定する必要があります。
-m start_time end_time	生成するレポートの時間範囲を指定します。開始時刻と終了時刻は、24 時間形式の時単位で指定する必要があります。
-u snapshot_specification	レポートで考慮する AWR（Automatic Workload Repository）スナップショットの範囲を指定します。
-b blacklist_file	ブラックリストとして使用する代替ファイルを指定します。
-o add-to-blacklist	デフォルトのブラックリストにファイルを追加します。
-o remove-from-blacklist	デフォルトのブラックリストからファイルを削除します。
-o print-blacklist	ブラックリストにあるファイルを表示します。
-a	Cached ODM が有効になっているファイルを、生成されるレポートに含めます。

オプション	用途
-p	表形式を無効化して、プログラム処理に適した出力を生成します。

SFHA 環境の Cached ODM Advisor を使用した候補データファイルのレポートの生成

dbed_codm_adm コマンドを使用することにより、Cached ODM を有効化するのに適したデータファイルを決定する助けとなるいくつかのレポートを生成できます。これらのレポートは、dbed_codm_adm コマンドの -o display オプションを使用して生成します。これらのレポートのほとんどは、データベースに有効化された定期的な AWR（自動作業負荷のリポジトリ）のスナップショットがあることを必要とします。

SFHA 環境の Cached ODM Advisor を使用した現在の読み取りアクティビティの基本レポートの生成

この基本のレポートはインスタンスが開始されてから起きた読み取りのアクティビティの概略を示します。レポートは Oracle によって提供される V$FILESTAT の表示で入手可能な情報の概略を示します。デフォルトでは、物理読み取り（PHYRDS）の最大数があった 10 のファイルが示されます。表示されるファイルの数は -n オプションを使用して変更でき、-c オプションを使用してカラムをソートできます。-t オプションを使用すると、レポートを特定の表領域に制限できます。

メモ: dbed_codm_adm コマンドは、システムの空きメモリがシステムメモリの合計の 25% 未満になった場合に警告を生成します。十分な空きメモリがない状態で Cached ODM を有効化すると、パフォーマンスに悪影響がある場合があります。

データベースに有効化された定期的な AWR のスナップショットがない場合は、生成できる種類はこのレポートが唯一のものです。

次はデータベースインスタンスが開始されて以来最大の読み取りのアクティビティがあったデータファイルのレポートが生成される例です。

```
$ dbed_codm_adm -S sfaedb -H $ORACLE_HOME -o display

File I/O statistics from Oracle V$FILESTAT view sorted on PHYRDS in descending order:
FILENAME                PHYRDS PHYWRTS PHYBLKRD PHYBLKWRT READTIM WRITETIM
-----------------------------------------------------------------------------

/data/sfaedb/data02.dbf 2225795 1453298 2291527 2087071 5371446 521421
/data/sfaedb/data03.dbf 2225697 1368342 2291338 2135174 9070509 552163
/data/sfaedb/data04.dbf 2156475 1327998 2221101 2054575 9036635 584703
```

```
/data/sfaedb/data05.dbf 2127539 1343213 2187749 2060534 8854222 626665
/data/sfaedb/data06.dbf 2076911 1306907 2135606 1972178 8600763 659874
/data/sfaedb/data07.dbf 2025810 1265414 2084207 1910216 8434117 667847
/data/sfaedb/data08.dbf 1937825 1171931 1996027 1779837 7991536 673212
/data/sfaedb/data09.dbf 1875733 1154502 1934094 1786911 7777172 692082
/data/sfaedb/data10.dbf 1819224 1114452 1877443 1721618 7521514 690706
/data/sfaedb/data01.dbf 1750829 1086044 1805138 1691258 7267005 357762
```

SFHA 環境の Cached ODM Advisor を使用した履歴アクティビティの要約レポートの生成

Cached ODM Advisor は、データファイルの履歴読み取りアクティビティのを要約するレポートを生成できます。これらのレポートは、データベースで定期的に作成される AWR スナップショットのデータを使用します。レポートを微調整するためにいくつかのオプションを使用できます。

次のいずれかのオプションを使用して生成されるレポートには次のカラムがあります。

- 表領域名
- ファイル名
- スナップショットごとの平均物理読み取り数
- スナップショットごとに読み取られる平均物理的ブロック数
- 読み取りの時間（秒単位）

-t オプションを使用してレポートを特定の表領域に制限できます。Cached ODM が有効化されているデータファイルはこのレポートには表示されません。それらのデータファイルもレポートに含める場合は、 -a オプションを使用します。デフォルトでは、上位 10 件のデータファイルがレポートで表示されます。このデフォルトは -n オプションを使用して変更できます。-p オプションは、デフォルトで実行される表のフォーマットを無効にすることにより、プログラムで解析可能なレポートを生成します。

Cached ODM Advisor ブラックリストを使用した SFHA 環境のファイルの無視

チューニングや測定によって特定のデータファイルで Cached ODM を有効化してもパフォーマンスの向上が得られないことが示唆される場合は、そのファイルをブラックリストに追加できます。ブラックリストに追加されるファイルは、dbed_codm_adm の -o display オプションを使用することで生成されるいずれのレポートにも含まれなくなります。

ブラックリストファイルは、lost+found/odmadm-blacklist のファイルシステムに格納されます。-b オプションを使用すると、ブラックリストに別の場所を提供することもできます。

-o add-to-blacklist および -o remove-from-blacklist のオプションを使用してデフォルトのブラックリストを管理できます。これらのオプションは単一のファイル名か、または -f filelist オプションを使用してファイルの一覧で使用できます。-o

print-blacklist オプションは、ブラックリストされたファイルの一覧を表示します。ブラックリストはレポートを生成するときのみのみ参照されます。データファイル上で Cached ODM を有効化しても無効化しても何の影響もありません。

SFHA 環境の Cached ODM Advisor を使用した時間間隔中の平均計測値

-d start_date end_date オプションと -m start_hour end_hour オプションを使うと、指定した時間間隔に記録されたアクティビティにレポートを制限できます。これには start_date および end_date パラメータが含まれています。したがって、この日付に起きるアクティビティも生成されたレポートに含まれます。start_hour と end_hour の両パラメータは、AWR のスナップショットを区切る時間間隔を日ごとに指定します。このレポートは、Cached ODM Advisor ユーティリティを初めて使用する場合で、ほとんどの読み取りアクティビティが作業負荷のかかる時間帯に集中しているデータファイルの一覧が必要な場合に役立ちます。

以下の例では、2011/07/09 から 2011/07/11 の午前 9 時から午後 5 時の間の DATA 表領域に対する読み取りアクティビティのレポートを生成します。

```
$ dbed_codm_adm -S sfaedb -H $ORACLE_HOME -o display ¥
 -d 2011/07/09 2011/07/11 -m 9 17 -t DATA
Tablespace  Filename                      Average   Average   Read
                                          Physical  Physical  Time
(s)
                                          Reads per Blocks
                                          Snapshot  Read per
                                                    Snapshot
DATA        /data/sfaedb/data03.dbf       13848     13848     497.16
DATA        /data/sfaedb/data04.dbf       13103     13104     472.68
DATA        /data/sfaedb/data05.dbf       12656     12656     451.79
DATA        /data/sfaedb/data06.dbf       12154     12154     430.81
DATA        /data/sfaedb/data07.dbf       11937     11937     423.87
DATA        /data/sfaedb/data08.dbf       11853     11853     417.74
DATA        /data/sfaedb/data09.dbf       11720     11720     415.45
DATA        /data/sfaedb/data10.dbf       11298     11299     400.59
DATA        /data/sfaedb/data01.dbf       10610     10611     390.25
Average per-block read time: 33.539 ms
```

SFHA 環境の Cached ODM Advisor を使用した AWR のスナップショットの指定した範囲の間隔の間の平均計測値

dbed codm_adm コマンドの -u snapshot_specification オプションを使用して、AWR のスナップショットの指定した範囲に記録されたアクティビティにレポートを制限できます。snapshot_specification は単一の整数にできますが、その場合はレポートは直近から開始してさかのぼる形で多くのスナップショットを考慮します。これにより最近読み込まれたアクティビティに関するレポートを生成できます。また、snapshot_specification

を使用して、AWR スナップショット id の範囲を starting_snapshot_id-ending_snapshot_id 形式で指定することもできます。これによりレポートは指定された範囲のスナップショット ID のスナップショットを考慮します。

次は、直近の 10 の AWR のスナップショットで指定された間隔の間に読み込まれたアクティビティのレポートを生成する例です。

```
$ dbed_codm_adm -S sfaedb -H $ORACLE_HOME -o display -u 10 -t DATA
Tablespace  Filename                    Average   Average   Read
                                        Physical  Physical  Time (s)

                                        Reads per Blocks
                                        Snapshot  Read per
                                                  Snapshot
DATA        /data/sfaedb/data06.dbf     31188     31188     1547.12
DATA        /data/sfaedb/data05.dbf     31131     31131     1542.57
DATA        /data/sfaedb/data03.dbf     30757     30757     1500.78
DATA        /data/sfaedb/data04.dbf     30687     30687     1525.45
DATA        /data/sfaedb/data07.dbf     30569     30569     1522.47
DATA        /data/sfaedb/data08.dbf     28781     28781     1422.77
DATA        /data/sfaedb/data09.dbf     27789     27789     1372.41
DATA        /data/sfaedb/data10.dbf     26828     26828     1317.54
DATA        /data/sfaedb/data01.dbf     24978     24978     1223.66
Average per-block read time: 46.449 ms
```

次は、AWR のスナップショット ID 80 から 90 までで指定された間隔の間に読み込まれたアクティビティのレポートを生成する例です。

```
$ dbed_codm_adm -S sfaedb -H $ORACLE_HOME -o display -u 80-90 -t DATA
Tablespace  Filename                    Average   Average   Read
                                        Physical  Physical  Time (s)

                                        Reads per Blocks
                                        Snapshot  Read per
                                                  Snapshot
DATA        /data/sfaedb/data03.dbf     38896     38896     1544.60
DATA        /data/sfaedb/data04.dbf     37649     37649     1586.97
DATA        /data/sfaedb/data05.dbf     37139     37139     1532.11
DATA        /data/sfaedb/data06.dbf     35875     35875     1462.15
DATA        /data/sfaedb/data07.dbf     34451     34451     1404.10
DATA        /data/sfaedb/data08.dbf     32219     32219     1284.36
DATA        /data/sfaedb/data10.dbf     31451     31451     1247.67
DATA        /data/sfaedb/data01.dbf     31377     31377     1254.35
DATA        /data/sfaedb/data09.dbf     31320     31320     1243.18
Average per-block read time: 37.691 ms
```

SFHA 環境の Cached ODM Advisor を使用した個々のデータファイルでの履歴アクティビティのレポートの生成

Cached ODM を有効化する対象としてのデータファイルを考慮する場合、個々のデータファイルのより詳細なレポートを取得できると便利です。そのようなレポートを生成するには dbed_codm_adm コマンドの -o display オプションをデータファイルに指定します。

このレポートには次のカラムがあります。

- スナップショット ID
- 間隔の終了日時
- 物理読み取り
- 物理ブロック読み取り
- 読み取りの時間（秒単位）

日付と時間範囲オプション (-d と -m) および dbed_codm_adm コマンドのスナップショットの範囲オプション -u を使用して、これらのレポートで考慮されるアクティビティを制御できます。さらに、個々のデータファイルのレポートを生成する場合に、スナップショット ID の複数の範囲を使用して比較レポートを生成できます。2 度めに異なる範囲のスナップショット ID で -u starting_snapshot_id-ending_snapshot_id オプションを指定して、2 つの時間間隔の間のデータファイルにおける読み取りのパフォーマンスを比較するレポートを生成できます。このレポートを使用して Cached ODM が有効化される前と後で読み取りのパフォーマンスを比較できます。

次は、データファイルで Cached ODM が有効化される前と後のデータファイル上での読み取りのパフォーマンスを比較するレポートを生成する例です。

```
$ dbed_codm_adm -S sfaedb -H $ORACLE_HOME -o display ¥
 -u 58-69 -u 70-80 /data/sfaedb/data02.dbf
Filename: /data/sfaedb/data02.dbf
Current ODM caching status: ON
Currently blacklisted: no

from AWR snapshot id 58 to 69
Snapshot  Interval End Time     Physical    Physical   Read Time
Id                              Reads       Blocks     (s)
                                            Read
58        2011/07/11 01:30:22   33026       33026      1188.13
59        2011/07/11 02:30:35   39158       39158      1358.33
60        2011/07/11 03:30:43   49186       49186      1694.08
61        2011/07/11 04:30:53   46941       46941      1652.38
62        2011/07/11 05:30:00   44090       44090      1519.78
63        2011/07/11 06:30:09   43095       43095      1511.17
```

```
64        2011/07/11 07:30:19   41272       41272      1430.41
65        2011/07/11 08:30:26   39869       39869      1379.52
66        2011/07/11 09:30:33   38540       38540      1362.99
67        2011/07/11 10:30:45   37157       37157      1309.31
68        2011/07/11 11:30:01   36034       36034      1250.8
69        2011/07/11 12:30:10   46344       46344      934.87
Average per-block read time: 33.538 ms

from AWR snapshot id 70 to 80
Snapshot  Interval End Time     Physical    Physical   Read Time
Id                              Reads       Blocks     (s)
                                            Read
70        2011/07/11 13:30:16   48041       48041      951.42
71        2011/07/11 14:30:24   46978       46978      809.31
72        2011/07/11 15:30:36   47077       47077      127.41
73        2011/07/11 16:30:47   44140       44140      370.01
74        2011/07/11 17:30:56   41467       41467      993.7
75        2011/07/11 18:30:10   39998       39998      867.62
76        2011/07/11 19:30:18   40812       40812      517.06
77        2011/07/11 20:30:29   38918       38918      706.19
78        2011/07/11 21:30:38   37572       37572      856.43
79        2011/07/11 22:30:53   37694       61675      594.25
80        2011/07/11 23:30:04   38048       38048      151.29
Average per-block read time: 14.327 ms
```

SFHA 環境の Cached ODM Advisor を使用したデータファイルの Cached ODM の有効化/無効化

Cached ODM Advisor を使用してデータファイルの Cached ODM 設定を管理することにより、odmadm 設定ファイルに手動で各ファイルを追加するよりも、管理性と簡便性を大幅に高めることができます。Cached ODM Advisor は、*mountpoint*/lost+found/odmadm 設定ファイルにツールによって設定される各データファイルのエントリを作成することで、Cached ODM 設定を永続的なものとします。必要に応じて、初回は設定ファイルを作成します。

Cached ODM Advisor を使用してデータファイルの Cached ODM 設定を管理する場合は、odmadm 設定ファイルを手動で編集しないでください。Cached ODM Advisor によりデータファイルに適用された設定は、/etc/vx/odmadm 設定ファイルで指定されている同じファイルのどの設定よりも優先されます。

Cached ODM Advisor を使用して、データベースのデータファイルで Cached ODM 設定を有効または無効にすることができます。

```
dbed_codm_adm -S ORACLE_SID -H ORACLE_HOME -o [ on | off ]  datafile
 | ¥
-f list_file
```

データベースのデータファイル上で Cached ODM を有効または無効にする場合は、以下の制限事項があります。

- `datafile` と `list_file` の値には、絶対パス名が含まれている必要があります。
- `dbed_codm_adm` コマンドでは、ファイルレベルでのみ Cached ODM を有効にします。

前提条件

- 指定したファイルがデータベースのデータファイルであること。そうでなければ、エラーが返されます。
- 指定したファイルが VXFS ファイルシステム上にあること。そうでなければ、エラーが返されます。

注意

- `dbed_codm_adm` は、I/O ヒントに基づいて Cached ODM を有効にせず、I/O ヒント情報を表示するだけです。
- データファイル名と `-f list_file` オプションのいずれかを指定する必要があります。

Cached ODM を有効化するには

◆ データファイル */oradata/tb1.dbf* で Cached ODM を有効化するには、次のコマンドを実行します。

```
$ dbed_codm_adm  -S prod -H /orahome -o on /oradata/tb1.dbf
```

データファイルのリストで Cached ODM を無効化するには

◆ */tmp/list_file* にデータファイル名のリストが含まれるデータファイルのリストで Cached ODM を無効化するには、次のコマンドを実行します。

```
$ dbed_codm_adm -S prod -H /orahome -o off -f /tmp/list_file
```

ユーザーがファイル上で Cache ODM を有効化しようとしたときに、ファイルごとのシステムフラグ odm_cache_enable が設定されていない場合、dbed_codm_adm は該当のファイルシステムに対して Cache ODM を自動的に有効化します。ファイルレベルのキャッシュを有効化できるように、このフラグを最初に有効化する必要があります。これには root アクセスが必要なので、DBED デーモンを通して実行されます。

キャッシュの有効化と無効化の制御の流れについて例を示します。

ファイル abc に対して Cached ODM を有効化する場合

- Cached ODM Advisor は、ファイルシステムの odm_cache_enable フラグがまだ有効になっていなければ、そのフラグを有効にします。
- Cached ODM Advisor は、odmadm setcachefile abc=on コマンドを実行し、設定ファイルにエントリを追加するか、更新します。

abc に対してキャッシュを無効化する場合

- Cached ODM Advisor は、odmadm setcachefile abc=off コマンドのみを実行し、設定ファイルのエントリも更新します。
- Cached ODM Advisor は、Cached ODM が有効なファイルが他に存在する可能性があるため、ファイルシステムの odm_cache_enable フラグを無効にしません。SFDB ツールには、ファイルシステムの odm_cache_enable フラグを無効化するコマンドはありません。ファイルシステムの odm_cache_enable フラグが他の方法によって無効化されている場合、ファイルごとのキャッシュは有効なままに見えますが、キャッシュは無効です。

SFHA 環境の Cached ODM Advisor を使用したデータベースファイルの Cached ODM 状態の表示

すべてのデータベースファイルとそれらの状態を表示するコマンドオプションを以下に示します。データファイル名 | -f list_file パラメータはオプションです。データファイル名または -f list_file パラメータが指定されていない場合、データベース内のすべてのファイルが表示されます。

```
dbed_codm_adm -S ORACLE_SID -H ORACLE_HOME -o filestate ¥
[ datafile | -f list_file ]
```

Cached ODM は 2 つのレベルで設定されます。

- ファイルシステムレベル
  ファイルシステムレベルでは、ON と OFF の 2 つの状態があります。
- ファイルレベル
  ファイルレベルでは、ON、OFF、DEF（デフォルト）の 3 つの状態があります。

表 11-2 に、最も効果的な Cached ODM の設定を決定する方法を示します。

表 11-2

		ファイルレベル	
ファイルシステムレベル	DEF	ON	OFF
ON	DEF	ON	OFF
OFF	OFF	OFF	OFF

dbed_codm_adm コマンドで表示されるのは、Cached ODM 設定の実際の状態です。存在しないファイルやファイルシステム以外のファイルでは、エラー状態（ERR）と呼ばれる特別な状態が表示されます。

Cached ODM が ON に設定されているファイルが先で、Cached ODM が OFF に設定されているファイルが後になるように、出力はソートされます。これで、Cached ODM が ON に設定されているファイルを見つけやすくなります。【★大量のファイルが表示される場合は、★】

ファイルの CODM 状態を表示するには

◆ */tmp/list* に存在するファイルの CODM 状態を表示するには、次のコマンドを実行します。

```
$ dbed_codm_adm -S prod -H /orahome -o filestate -f /tmp/list

Effective Cached ODM Status For Datafiles

(State value 'ERR' implies non-vxfs or non-existent file.)
FILENAME                                                STATE
-------------------------------------------------------------
/data/CODM/system01.dbf                                  DEF
/data/CODM/undotbs01.dbf                                 DEF
/data/CODM/sysaux01.dbf                                  DFF
/data/CODM/users01.dbf                                   OFF
/data/CODM/example01.dbf                                 ON
```

SFHA 環境の Cached ODM Advisor を使用した Cached ODM 統計情報の表示

データベースファイルで Cached ODM が有効化された後は、ODM 統計情報を表示することで、Cached ODM をオンにする有効性を確認することができます。ODM 統計情報には、読み取り操作と書き込み操作の回数、読み取りと書き込みのブロック数、キャッシュのヒット率などが含まれます。ユーザーは、ヒット率を確認することで、Cached ODM がどの程度有効であるかについて何らかのヒントを得ることができます。

前提条件

- 指定したファイルがデータベースのデータファイルであること。そうでなければ、エラーが返されます。
- 指定したファイルが VXFS ファイルシステム上にあること。そうでなければ、エラーが返されます。

注意

- 個別のデータファイルまたはデータファイルのリストが指定されていないと、指定した ORACLE_SID で Cached ODM が有効になっているすべてのファイルの odmstats が表示されます。
- ファイルが開かれなくなると、統計情報は 0 にリセットされます。
- ODM 統計情報がファイルシステムで収集されるのは、odm_cache_enable が設定されている場合のみです。データファイルを含むファイルシステムで odm_cache_enable が設定されていない場合、CACHE STATISTICS や HIT RATIO などの情報は 0 に設定されます。

1 つのデータベースファイルの ODM 統計情報を表示するコマンドオプションを次に示します。データファイル名の指定または–f list_file オプションの使用が必須です。

```
dbed_codm_adm -S ORACLE_SID -H ORACLE_HOME -o odmstats [  datafile | -f list_file ]
```

たとえば、/oradata/tb1.dbf データファイルの ODM 統計情報を表示するには、次のコマンドを使います。

```
$ dbed_codm_adm -S prod -H /orahome -o odmstats /oradata/tb1.dbf


FILENAME        NREADS NWRITES RBLOCKS WBLOCKS RTIME WTIME NREQUESTIO NDISKIO HIT-RATIO

-----------------------------------------------------------------------------------------
/oradata/tb1.dbf 1262    10617   51040   287056  11.0  1.0        4728    5918     22.0
```

SFHA 環境の Cached ODM Advisor を使用した ODM I/O 統計情報の表示

ODM I/O 統計情報を表示することができます。I/O 統計情報には、各ファイルタイプと I/O タイプの組み合わせについて、I/O 要求や処理済みデータ量などの情報が含まれます。

ローカルホストの ODM I/O 統計情報を表示するには、次のコマンドを使用します。

```
$ dbed_codm_adm -o iostats
```

ODM I/O 統計情報を表示するには、以下の点を考慮します。

- 表示される I/O 統計情報はローカルノードに固有であり、すべてのファイルシステムにわたるため、-S ORACLE_SID、-H ORACLE_HOME、データファイル、-f list_file の各オプションは必須ではありません。
- Oracle RAC クラスタで使用されている場合、表示される統計情報はすべてのノードから集められた統計情報ではありません。
- VxFS の odmadm getiostats コマンドを内部的に使って、データをフィルタ処理します。
- I/O 統計情報は操作の回数を示すのみで、操作がキャッシュに保存されたかどうかを示すものではありません。
- ODM I/O 統計情報がファイルシステムで収集されるのは、odm_cache_enable が設定されている場合のみです。

ODM I/O 統計情報を表示するには、次のコマンドを使用します。

```
$ dbed_codm_adm -o iostats


FILE-TYPE/IO-TYPE               OPERATIONS        BYTES
----------------------------------------------------------------
data/data_cache_load            33                270336
data/read_data_header           104               851968
data/data_read_seq              2                 16384
olg/redolog_read                98                98283008
```

12

Quick I/O によるデータベースパフォーマンスの向上

この章では以下の項目について説明しています。

- Quick I/O について
- qiomkfile による Quick I/O ファイルとしての Oracle データベースファイルの作成
- setext コマンドによる Quick I/O ファイル領域の事前割り当て
- 通常の VxFS ファイルへの Quick I/O ファイルとしてのアクセス
- Oracle ファイルの Quick I/O ファイルへの変換
- スパースファイルについて
- Oracle データベースにおける一時表領域と Quick I/O の扱い
- Quick I/O の状態とファイル属性の表示
- Quick I/O ファイルの拡張
- Quick I/O ファイルでの Oracle AUTOEXTEND の使用
- データベースのリストア後の Quick I/O ファイルの再作成
- Quick I/O の無効化
- Solaris ローカルゾーンへの Quick I/O ファイルの作成

Quick I/O について

Veritas Quick I/O は、Veritas InfoScale Storage Foundation Standard 製品と Enterprise 製品に含まれる VxFS 機能です。この機能によって、アプリケーションは事前

に割り当てられた VxFS ファイルを RAW デバイスとして扱い、そのファイルにアクセスできます。Quick I/O 機能を使うと、ファイルシステムでデータベースを実行するうえで管理上のメリットがあります。通常発生するデータベース実行に関連するパフォーマンスの低下はありません。

メモ: Quick I/O は Linux ではサポートされません。

Quick I/O によるデータベースパフォーマンス向上のしくみ

Quick I/O を使う利点は次のとおりです。

- Quick I/O ファイルを RAW デバイスとして機能させることにより、パフォーマンスと処理スループットが向上します。
- Quick I/O ファイルを通常のファイルとして管理できます。これにより、Oracle データファイルの割り当て、移動、コピー、サイズ変更およびバックアップなどの管理作業を簡単に行うことができるようになります。

  メモ: Veritas は、Oracle Disk Manager の使用を推奨します。

メモ: Quick I/O は Linux ではサポートされません。

通常のファイルを RAW デバイスとして扱ってアクセスする Quick I/O の機能により、次のことが実現され、データベースのパフォーマンスが向上します。

表 12-1

Quick I/O 機能	利点
ダイレクト I/O のサポート	`read()` および `write()` システムコールを使うファイルに対する I/O は、通常、データを 2 回コピーします。1 回目はユーザーとカーネル領域の間で、2 回目はカーネル領域とディスクの間で実行されます。これとは異なり、RAW デバイス上の I/O は直接処理されます。すなわち、データをユーザー領域とディスクの間で直接コピーして、コピーの手順を 1 つ省略します。RAW ディスク上の I/O と同様に、Quick I/O でも余分なコピーは実行しません。

Quick I/O 機能	利点
データベースファイルに対するカーネル書き込みロックの回避	データベース I/O を write() システムコールを介して処理する場合、各システムコールは、カーネル内部で書き込みロックを取得し、解放します。このロックにより、同一ファイルで同時に複数の書き込み操作が実行されるのを防ぎます。データベースシステムには、通常、ファイルへの並行アクセスを管理するために独自のロック機能が実装されているため、ファイル単位の書き込みロックが不必要に I/O 処理をシリアル化してしまいます。Quick I/O は、このファイルシステムに対するファイル単位のロックを回避し、データベースサーバーがデータアクセスを制御できるようにします。
二重バッファリングの回避	ほとんどのデータベースサーバーは、独自のバッファキャッシュを持っているため、システムのバッファキャッシュを必要としません。そのため、ファイルシステムバッファにキャッシュされたデータベースのデータは冗長であり、メモリ領域やバッファ管理のための CPU 使用率の浪費になります。Quick I/O がダイレクト I/O をサポートすることにより、二重バッファリングが不要になります。データはリレーショナルデータベース管理システム（RDBMS）のキャッシュとディスクの間で直接コピーされるため、CPU 使用率の低下とメモリの解放につながり、データベースサーバーのバッファキャッシュが使用できるメモリ量が増加することにより、トランザクション処理のスループットが向上します。
AIX の場合: AIX Fastpath 非同期 I/O のサポート	AIX Fastpath 非同期 I/O は、ノンブロッキングのシステムレベルの読み取りおよび書き込みを実行することによって複数の I/O 要求の同時処理を可能にする I/O 方法です。AIX などのオペレーティングシステムでは、RAW デバイスに対する非同期 I/O がサポートされていますが、標準ファイル I/O に対してはサポートされていません。そのため、データベースサーバーで非同期 I/O を使用できる場合でも、データベースがファイルシステム上で動作するときには非同期 I/O 要求を発行できません。非同期 I/O を行うことができないと、パフォーマンスが大幅に低下します。Quick I/O を使用すると、データベースサーバーは、Quick I/O インターフェースを使用してアクセスするファイルシステム上のファイルに対してカーネルでサポートされる非同期 I/O を利用できます。

Quick I/O 機能	利点
Solaris の場合: カーネルの非同期 I/O のサポート	Solaris のカーネル非同期 I/O は、ノンブロッキングのシステムレベルの読み取りおよび書き込みを実行することによって複数の I/O 要求の同時処理を可能にする I/O 方法です。Solaris などのオペレーティングシステムでは、RAW デバイスに対するカーネルの非同期 I/O がサポートされていますが、標準ファイル I/O に対してはサポートされていません。そのため、データベースサーバーで非同期 I/O を使用できる場合でも、データベースがファイルシステム上で動作するときには非同期 I/O 要求を発行できません。非同期 I/O を行うことができないと、パフォーマンスが大幅に低下します。Quick I/O を使用すると、データベースサーバーは、Quick I/O インターフェースを使用してアクセスするファイルシステム上のファイルに対してカーネルでサポートされる非同期 I/O を利用できます。

qiomkfile による Quick I/O ファイルとしての Oracle データベースファイルの作成

qiomkfile は、表領域コンテナに領域を事前に割り当て、Quick I/O インターフェースを通してアクセスできるようにする最良の方法です。qiomkfile を使うと、一時表領域または通常の表領域に Quick I/O ファイルを作成できます。

メモ: Quick I/O は Linux ではサポートされません。

前提条件

- Quick I/O ファイルを作成できるのは、VxFS ファイルシステムのみです。
- 既存のファイルシステム上にデータベースファイルを作成する場合は、fsadm または同様のユーティリティを実行して断片化のレポートと解消を行ってください。
- Oracle の Quick I/O ファイルを作成するには、作成先のディレクトリに対する読み取り権限と書き込み権限が必要です。

使用に関する注意事項

- qiomkfile コマンドにより、事前に割り当てられた連続領域を含む標準ファイルと、Quick I/O の名前拡張子を示すシンボリックリンクの 2 つのファイルが作成されます。
- 詳しくは qiomkfile(1M) のマニュアルページを参照してください。

-a

指定されたファイルの絶対パス名を使ってシンボリックリンクを作成します。-a オプションは、絶対パス名が必要な場合に使います。ただし、デフォルトでは、相対パス名でシンボリックリンクが作成されます。

-e
: Oracle 表領域のサイズ変更を可能にするため、ファイルを指定量だけ拡張します。

p.133 の「Quick I/O ファイルの拡張」を参照してください。

-h
: Oracle データファイルのヘッダーサイズを指定します。このオプションでは指定したサイズ以外に割り当てるヘッダーを指定します。Oracle では、すべてのデータファイル用のデータベースブロックが追加で 1 つ必要となるためです。このオプションを使った場合、作成したファイルを Oracle データファイルとして使えます。Oracle データファイルを作成する場合、ヘッダーサイズは（DB_BLOCK_SIZE パラメータで指定した）Oracle ブロックサイズと同じにする必要があります。-h オプションを使うときにヘッダーサイズを指定しなかった場合は、32 K のヘッダーが割り当てられます。

-r
: Oracle 表領域のサイズ変更を可能にするため、ファイルを指定サイズに拡張します。

p.133 の「Quick I/O ファイルの拡張」を参照してください。

-s
: AIX と Solaris の場合:

ファイルに事前に割り当てる領域を指定します。k、K、m、M、g、G、s、S のいずれかの接尾辞を追加することで、バイト単位、KB 単位、MB 単位、GB 単位、セクタ（512 バイト）単位のいずれかで指定できます。

デフォルトでは、バイト単位が使われます。バイト単位で値を指定する場合は、接尾辞を付加する必要はありません。事前に割り当てるファイルのサイズは、（ヘッダーを含む）ファイルの合計サイズが、ファイルシステムのブロックサイズの倍数に切り上げられます。

警告: 絶対パス名を使う場合には、十分な注意が必要です。シンボリックリンクを維持するために、データベースのバックアップとリストア時に追加手順が必要になることがあります。もとのパスと異なるディレクトリにファイルをリストアする場合、データベースを再起動する前に、新しいパス名を指す絶対パス名を使うようにシンボリックリンクを変更してください。

---

qiomkfile を使って Quick I/O ファイルとしてデータベースファイルを作成するには

1. qiomkfile コマンドを使って、データベースファイルを作成します。

```
# /opt/VRTS/bin/qiomkfile -h headersize -s file_size
/mount_point/filename
```

2. ファイルの所有者およびグループのアクセス権を変更します。

```
# chown oracle:dba .filename

# chmod 660 .filename
```

3. SQL*Plus 文を使い、このファイル上に表領域を作成します。

   次に例を示します。

```
$ sqlplus /nolog

SQL> connect / as sysdba

SQL> create tablespace ts1 datafile '/mount_point/filename.dbf'

size 100M reuse;

exit;
```

次は、相対パス名を使って、VxFS ファイルシステム /db01 上に dbfile という 100 MB のデータベースファイルを作成する例です。

```
# /opt/VRTS/bin/qiomkfile -h 32k -s 100m /db01/dbfile

# ls -al
```

AIX の場合:

```
-rw-r--r--   1 oracle   dba   104857600   Oct 2 13:42    .dbfile
```

Solaris の場合:

```
-rw-r--r--   1 oracle   dba   104890368   Oct 2 13:42    .dbfile

lrwxrwxrwx   1 oracle   dba          19   Oct 2 13:42    dbfile ->
¥
.dbfile::cdev:vxfs:
```

この例では、qiomkfile によって /db01/.dbfile という通常のファイルが作成され、実領域が割り当てられています。次に、qiomkfile によって /db01/dbfile というシンボリックリンクが作成されます。このシンボリックリンクは、/db01/.dbfile の Quick I/O イ

ンターフェースへの相対リンク、つまり .dbfile::cdev:vxfs: ファイルへの相対リンクです。このシンボリックリンクにより、任意のデータベースまたはアプリケーションから Quick I/O インターフェースを使って .dbfile にアクセスできるようになります。

setext コマンドによる Quick I/O ファイル領域の事前割り当て

qiomkfile コマンドの他、VxFS の setext コマンドを使っても、データベースファイルに領域を事前に割り当てることができます。

setext で領域を事前に割り当てる前に、次の条件が満たされていることを確認してください。

前提条件
- setext コマンドには、スーパーユーザー（root）権限が必要です。

使用に関する注意事項
- chown コマンドを使えば、作成したファイルの所有者およびグループのアクセス権を変更できます。
  詳しくは setext（1M）のマニュアルページを参照してください。

setext を使って Quick I/O データベースファイルを作成するには

1. VxFS のマウントポイントにアクセスし、ファイルを作成します。

```
# cd /mount_point

# touch .filename
```

2. setext コマンドを使って、ファイルに領域を事前に割り当てます。

```
# /opt/VRTS/bin/setext -r size -f noreserve -f chgsize  ¥
.filename
```

3 シンボリックリンクを作成して、データベースまたはアプリケーションから Quick I/O インターフェースを通してファイルにアクセスできるようにします。

```
# ln -s .filename::cdev:vxfs: filename
```

4 ファイルの所有者およびグループのアクセス権を変更します。

AIX の場合:

```
# chown user:group .filename
```

Solaris の場合:

```
# chown oracle:dba .filename

# chmod 660 .filename
```

次は、Oracle のマウントポイント /db01 にアクセスし、データファイルを作成し、領域を事前に割り当て、アクセス権を変更する例です。

```
# cd /db01
# touch .dbfile
# /opt/VRTS/bin/setext -r 100M -f noreserve -f chgsize .dbfile
# ln -s .dbfile::cdev:vxfs: dbfile

# chown oracle:dba .dbfile

# chmod 660 .dbfile
```

通常の VxFS ファイルへの Quick I/O ファイルとしてのアクセス

::cdev:vxfs: 名前拡張子を使うと、通常の VxFS ファイルに Quick I/O ファイルとしてアクセスできます。

シンボリックリンクを使うと、容易にファイルシステムを管理でき、場所を意識しないでデータベースファイルにアクセスできます。ただし、データベースのバックアップやリストア時などに管理が必要となるファイルが 2 セットになります。

使用に関する注意事項

- 通常のファイルに Quick I/O ファイルとしてアクセスするためのシンボリックリンクを作成するときは、可能であれば、絶対パス名ではなく、相対パス名を使ってください。相対パス名を使うと、ディレクトリをコピーした場合に、シンボリックリンクのコピーから最初のファイルが参照されるのを避けることができます。これは、シンボリックリンクを保存するコマンドでバックアップやデータベースファイルの移動を実行する場合に重要です。

  ただし、絶対パス名を必要とするアプリケーションもあります。この場合、ファイルを別のディレクトリに再配置するときは、新しい絶対パスを使うようにシンボリックリンクを変更する必要があります。または、データディレクトリとは別のディレクトリに、すべてのシンボリックリンクを格納することもできます。たとえば、/database という名前のディレクトリを作成し、そのディレクトリにすべてのシンボリックリンクを配置して、シンボリックリンクが絶対パス名を指すように指定できます。

VxFS ファイルシステム上の既存の通常ファイルに Quick I/O ファイルとしてアクセスするには

1. 通常のファイルが保持される VxFS ファイルシステムのマウントポイントにアクセスします。

```
$ cd /mount_point
```

2. シンボリックリンクを作成します。

```
$ mv filename .filename
$ ln -s .filename::cdev:vxfs: filename
```

これは、VxFS ファイル dbfile に Quick I/O ファイルとしてアクセスする例です。

```
$ cd /db01
$ mv dbfile .dbfile
$ ln -s .dbfile::cdev:vxfs: dbfile
```

これは、シンボリックリンクが作成されたことを確認する例です。

```
$ ls -lo .dbfile dbfile
```

AIX の場合:

```
lrwxrwxrwx  1 oracle      19     Oct 2 13:42   dbfile ->
.dbfile::vxcdev:vxfs:
```

Solaris の場合:

```
lrwxrwxrwx  1 oracle      19     Oct 2 13:42   dbfile ->
.dbfile::cdev:vxfs:
```

Oracle ファイルの Quick I/O ファイルへの変換

/opt/VRTSdbed/bin ディレクトリには、既存のデータベースを Quick I/O を使うように変換するための特別なコマンドがあります。qio_getdbfiles コマンドを使うと、データベースシステムテーブルからファイル名のリストを抽出でき、さらに qio_convertdbfiles コマンドを使うと、Quick I/O を使うようにこのリストのデータベースファイルを変換できます。

メモ: Storage Checkpoint は、Quick I/O に変換する前に作成することをお勧めします。

Quick I/O ファイルにデータベースを変換する前に、次の条件を満たす必要があります。

前提条件

- データベース管理者（一般的なユーザー ID は oracle）としてログインし、qio_getdbfiles および qio_convertdbfiles コマンドを実行します。
- 事前に Oracle 環境変数 $ORACLE_SID を定義しておく必要があります。環境変数 ORACLE_SID への変更が定義されている必要があります。
- 変換するファイルは、VxFS ファイルシステム上の通常のファイルまたは通常の VxFS ファイルを指すリンクである必要があります。

使用に関する注意事項

- 既存のデータベースファイルが断片化されている場合は、Quick I/O ファイルの変換が最良の結果をもたらさないことがあります。5.1 では、-f オプションを使った断片化レベルの指定はサポートされていません。
- 新しいファイルを作成すると、このファイルは連続したファイルになります。その後、dd(1M)コマンドまたはデータベースインポート機能を使用して、古いファイルから新しいファイルにデータを移動する必要があります。移動したら、データベースに新しいファイルを定義します。
- デフォルトでは、qio_getdbfiles は、TEMPORARY としてマーキングされたすべての表領域をスキップします。TEMPORARY とマーキングされた表領域は、スパース、すなわち割り当てられていないブロックがファイル内に存在する可能性があります。Quick I/O はストレージに対する RAW タイプのインターフェースなので、スパースファイルは Quick I/O ファイルに使えません。スパースファイルを Quick I/O ファイルに変換した場合、割り当てられていないブロックに Oracle が書き込もうとすると Oracle インスタンスの処理が失敗します。
  p.130 の「Oracle データベースにおける一時表領域と Quick I/O の扱い」を参照してください。
- また、一時表領域用に Quick I/O ファイルを作成することもできます。
  p.118 の「qiomkfile による Quick I/O ファイルとしての Oracle データベースファイルの作成」を参照してください。
- qio_convertdbfiles コマンドは、指定されたデータベースファイルが VxFS ファイルシステム上に存在しない場合、終了し、エラーメッセージを出力します。この場合、qio_convertdbfiles コマンドを実行する前に、mkqio.dat ファイルから VxFS ファイルシステム上に存在しないファイルを削除する必要があります。
- Quick I/O ファイルに変換する Oracle データベースのファイル名が含まれる mkqio.dat ファイルは、qio_getdbfiles コマンドを使わないで、手動で作成することもできます。

qio_getdbfiles コマンドの使用法は次のとおりです。

```
qio_getdbfiles SFUA qio_getdbfiles ERROR V-81-4265
```

qio_getdbfiles コマンドでは、次のオプションが使えます。

-a　　潜在的なスパースファイルを含むすべてのデータファイルを抽出します。

（スパースファイルは Quick I/O の対象外なので、このオプションはデバッグに限定して使ってください。）

-T　　データベースのタイプを ora として指定できます。

qio_getdbfiles コマンドの使用法は次のとおりです。

```
qio_getdbfiles [ -T database_type ] [ options ]
```

`qio_convertdbfiles` コマンドでは、次のオプションが使えます。

`-a` 絶対パス名を使って、通常のファイルを Quick I/O ファイルに変更します。このオプションは、シンボリックリンクが絶対パス名を指す必要がある場合（たとえば、SAP を使うサイト）に使います。

`-h` ヘルプメッセージを表示します。

SAP の `brbackup` をサポートするため、`/dev` ディレクトリ内のすべてのデータファイルおよびログファイルの追加リンクを作成します。

`-T` データベースのタイプを `ora` として指定できます。

`-u` Quick I/O ファイルを通常のファイルに戻します。このオプションは、前に実行した `qio_convertdbfiles` スクリプトによる変更をもとに戻すときに使います。

変換する Oracle データファイルのリストを抽出するには

◆ データベースインスタンスの実行中に、書き込み権限のあるディレクトリから qio_getdbfiles コマンドを実行します。

```
$ cd /extract_directory

$ /opt/VRTSdbed/bin/qio_getdbfiles -T ora
```

qio_getdbfiles コマンドにより、データベースシステムテーブルからファイル名の一覧が抽出され、そのファイル名とサイズ（バイト）が mkqio.dat というファイルに格納されます。

メモ: または、Quick I/O を使うために変換する Oracle データベースファイル名を含む mkqio.dat ファイルを手動で作成することもできます。また、qio_getdbfiles で作成した mkqio.dat ファイルを手動で編集したり、Quick I/O ファイルに変換しないファイルを削除することもできます。

メモ: qio_getdbfiles コマンドを実行するには、データベースへのアクセス権と /extract_directory への書き込み権限が必要です。

---

mkqio.dat リストファイルは、次のような形式になります。

```
/data11r1/VRTS11r1/redo01.log 52428800
/data11r1/VRTS11r1/redo02.log 52428800
/data11r1/VRTS11r1/redo03.log 52428800
/data11r1/VRTS11r1/sysaux01.dbf 632553472
/data11r1/VRTS11r1/system01.dbf 754974720
/data11r1/VRTS11r1/undotbs01.dbf 47185920
/data11r1/VRTS11r1/users01.dbf 5242880
/data11r1/nqio1.dbf 104857600
```

Oracle データベースファイルを Quick I/O ファイルに変換するには

1. データベースを停止します。
2. mkqio.dat ファイルを含むディレクトリで qio_convertdbfiles コマンドを実行します。

   ```
   $ cd /extract_directory
   ```

   ```
   $ /opt/VRTSdbed/bin/qio_convertdbfiles
   ```

   mkqio.dat ファイル内のファイルリストが表示されます。次に例を示します。

   ```
   file1 --> .file1::cdev:vxfs:
   file2 --> .file2::cdev:vxfs:
   file3 --> .file3::cdev:vxfs:
   file4 --> .file4::cdev:vxfs:
   file5 --> .file5::cdev:vxfs:
   ```

   qio_convertdbfiles コマンドの実行（オプションの指定なし）により、ファイル名が *filename* から .filename に変更され、Quick I/O 固有の拡張子を持つ .filename へのシンボリックリンクが作成されます。デフォルトでは、シンボリックリンクでは相対パス名が使われます。

   qio_convertdbfiles スクリプトは、指定されたデータベースファイルが VxFS ファイルシステム上に存在しない場合、終了し、エラーメッセージを出力します。この場合、qio_convertdbfiles コマンドを再度実行する前に、mkqio.dat ファイルから VxFS ファイルシステム上に存在しないすべてのファイルを削除する必要があります。
3. データベースを起動します。
4. これらのデータベースファイルに Quick I/O インターフェースを使ってアクセスできるようになりました。

以前に実行した qio_convertdbfiles を取り消して、Quick I/O ファイルを通常のファイルに戻すには、次のコマンドを実行します。

1 データベースが実行中の場合は、停止します。

2 `mkqio.dat` ファイルを含むディレクトリで次のコマンドを実行します。

```
$ cd /extract_directory

$ /opt/VRTSdbed/bin/qio_convertdbfiles -u
```

`mkqio.dat` ファイル内の Quick I/O ファイルのリストが表示されます。次に例を示します。

```
.file1::cdev:vxfs: --> file1
.file2::cdev:vxfs: --> file2
.file3::cdev:vxfs: --> file3
.file4::cdev:vxfs: --> file4
.file5::cdev:vxfs: --> file5
```

変換を元に戻すオプション（`-u`）が指定された `qio_convertdbfiles` コマンドを使うと、ファイルの名前は <.filename> から <filename> に変更され、Quick I/O ファイルとともに作成された `.filename` へのシンボリックリンクは元に戻されます。

スパースファイルについて

スパースファイルをサポートすると、アプリケーションが格納した情報（i ノード内部）から、ゼロのみが保持されるデータブロックを特定できるようになります。これにより、ゼロ以外のデータが保持されるブロックのみをディスク上に割り当てることができます。

たとえば、10 KB のファイルでは、一般に、ディスク上のブロックに 10 KB すべてが展開されます。ここでたとえば、最初の 9 K を常にゼロにしたいとします。アプリケーションは、オフセット 9 KB の位置から 1 KB 分のデータを書き込むことができます。割り当てられるブロックは書き込まれた 1 KB 分のみですが、ファイルのサイズは 10 KB のままです。

今、このファイルはスパースファイルになっています。このファイルのオフセット 0 から 9 KB まではホールになります。アプリケーションがこの範囲にあるデータを読み込むと、ゼロの文字列が取得されます。

その後、アプリケーションが 1 KB のブロックをオフセット 4 KB から書き込もうとすると、ファイルシステムは別のブロックを割り当てます。

これにより、ファイルは次のようになります。

- 0 - 4 KB - ホール
- 4 - 5 KB - データブロック
- 5 - 9 KB - ホール

- 9 - 10 KB - データブロック

そのため、ゼロのみを含むブロックの数が十分に多ければ、1 TB のファイルシステムで最大 2 TB 分のファイルを格納できます。Quick I/O ファイルはスパースにできません。指定されたすべてのブロックが必ず割り当てられます。

Oracle データベースにおける一時表領域と Quick I/O の扱い

Quick I/O ファイルを使って、新しい一時表領域を作成できます。ただし、Oracle9 では `qio_getdbfiles` コマンドを使って、通常のファイルを使う既存の一時表領域を Quick I/O に変換することはできません。

デフォルトでは、`qio_getdbfiles` は、`TEMPORARY` とマーキングされたすべての表領域を一覧に抽出しません。これは、ファイル内に割り当てられていないブロックが存在するスパースファイルである可能性があるためです。Quick I/O はストレージに対する RAW タイプのインターフェースなので、スパースファイルは Quick I/O ファイルに使えません。スパースファイルを Quick I/O ファイルに変換した場合、割り当てられていないブロックに Oracle が書き込もうとすると Oracle インスタンスの処理が失敗します。ただし、Quick I/O ファイル上に一時表領域を初めて作成したときは、Oracle はそれらを RAW デバイスと見なすため、スパースファイルを作成しません。

通常ファイルを使う一時表領域を Quick I/O ファイルに変換する場合、通常ファイルを使う既存の一時表領域を一旦、破棄すると、Quick I/O ファイルを使う一時表領域を再作成できます。一時表領域を通常ファイルとして残しておくこともできます。

一時表領域以外の表領域に使われているファイル名の一覧を取得するには

◆ 次の SQL 文を使います。

```
$ sqlplus /nolog
 SQL> connect / as sysdba;
 SQL> select file_name from dba_data_files a,
 dba_tablespaces b where a.tablespace_name =
 b.tablespace_name and b.contents <> 'TEMPORARY';
```

既存の一時表領域を破棄し、Quick I/O ファイルとして再作成するには

1. 一時表領域を内容とともに破棄します。

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> drop tablespace tablespace_name including contents;
```

2. VxFS ファイルシステムで Quick I/O ファイルを作成します。

```
# /opt/VRTS/bin/qiomkfile -h header_size -s size ¥
/mount_point/filename.dbf
```

3. ファイルの所有者およびグループのアクセス権を変更します。

```
# chown oracle:dba .filename

# chmod 660 .filename
```

4. `create temporary tablespace` コマンドを使って新しい一時表領域を作成します。

   `create temporary tablespace` コマンドの使用方法は、次のとおりです。

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> create temporary tablespace tablespace_name ¥
tempfile '/mount_point/new_filename.dbf' size size reuse;
```

この例では、表領域 `tempts` を破棄し、Quick I/O ファイル `temp01.dbf` を作成し、アクセス権限を変更してから、`create temporary tablespace` コマンドを使って、新しい一時表領域 `tempts` を作成する方法を示します。

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> drop tablespace tempts including contents;
Tablespace dropped.
# /opt/VRTS/bin/qiomkfile -h 32k -s 100M /db01/temp01.dbf

# chown oracle:dba .temp01
# chmod 660 .temp01

$ sqlplus /nolog
SQL> connect / as dba;
SQL> create temporary tablespace tempts ¥
tempfile '/db01/temp01.dbf' ¥
```

```
size 100M reuse;
Tablespace created.
```

Quick I/O の状態とファイル属性の表示

ls コマンドのさまざまなオプションを使うと、Quick I/O の状態とファイル属性に関する情報を取得し、表示できます。

-al	Quick I/O ファイルおよびそのリンクを含むファイルシステム上のすべてのファイルのリストを表示します。
-1L	Quick I/O が正常にインストールされ、有効にされているかどうかを表示します。
-a1L	Quick I/O ファイル名が RAW デバイスでどのように名前解決されているかを表示します。

Quick I/O ファイルとそのリンクを含む現在のファイルシステム上のすべてのファイルのリストを表示するには

◆ ファイル名を指定して、ls -al コマンドを使います。

```
$ ls -al filename .filename
```

次の例で、-a オプションを使って、qiomkfile で作成された絶対パス名を表示する方法を示します。

```
$ ls -la /snap_data11r2/FLAS11r2/.qfile01.dbf /snap_data11r2/FLAS11r2/qfile01.dbf
lrwxrwxrwx  1 ora11gr2  dba  24 Jul 15 03:27 /snap_data11r2/FLAS11r2/.qfile01.dbf -> ¥
.qfile01.dbf::cdev:vxfs:
lrwxrwxrwx  1 ora11gr2  dba  48 Jul 15 05:17 /snap_data11r2/FLAS11r2/qfile01.dbf -> ¥
/snap_data11r2/FLAS11r2/.qfile01.dbf::cdev:vxfs:
```

Oracle データファイルが Quick I/O に変換されたかどうかを判断するには

◆ 次のように、ls コマンドを使います。

```
$ ls -lL filename
```

次の例で、Quick I/O がインストールされ有効にされているかどうかを判断する方法を示します。

```
$ ls -lL dbfile

crw-r--r--  1 oracle  dba   45, 1        Oct 2 13:42  dbfile
```

AIX の場合:

上記で、最初の 1 文字 c は RAW キャラクタデバイスファイルであることを示しています。メジャーデバイス番号とマイナーデバイス番号がサイズのフィールドに表示されます。No such file or directory というメッセージが表示された場合は、Quick I/O が正しくインストールされていないか、有効なライセンスキーがありません。

解決された Quick I/O ファイルを表示するには

◆ 次のように、ファイル名を指定して ls コマンドを使います。

```
$ ls -alL filename .filename
```

次の例で、Quick I/O ファイル名 dbfile を RAW デバイスの名前で解決する方法を示します。

AIX の場合:

```
crw-r--r--   1 oracle   dba      45,  1        Oct 2 13:42
dbfile

$ ls -alL d* .d*
```

Solaris の場合:

```
crw-r--r--   1 oracle   dba      45,  1        Oct 2 13:42
dbfile
```

Quick I/O ファイルの拡張

Quick I/O ファイルは事前に割り当てられている必要がありますが、そのサイズは事前に割り当てられたサイズに制限されません。qiomkfile コマンドのオプションを使うと、Quick I/O ファイルを指定の量または指定のサイズに拡張できます。Quick I/O ファイルの拡張は、オンライン操作で迅速に行うことができ、RAW デバイスを使うよりはるかに有利です。

Quick I/O ファイルを拡張する前に、次の条件が満たされていることを確認してください。

前提条件

- Quick I/O ファイルを拡張するには、ファイルシステム上に十分な容量が必要です。

使用に関する注意事項

- fsadm コマンドを使って、オンラインで VxFS ファイルシステムを拡張することもきます（下位のディスクまたはボリュームが拡張可能な場合）。vxresize コマンドを使えば、下位のボリュームやファイルシステムを拡張できます。
- fsadm コマンドを使って VxFS ファイルシステムのサイズを変更するには、スーパーユーザー（root）権限が必要です。
- 詳しくは fsadm_vxfs（1M）および qiomkfile（1M）のマニュアルページを参照してください。

qiomkfile コマンドには次のオプションがあります。

-e	Oracle 表領域のサイズ変更を可能にするため、ファイルを指定量だけ拡張します。
-r	Oracle 表領域のサイズ変更を可能にするため、ファイルを指定サイズに拡張します。

Quick I/O ファイルを拡張するには

1. 必要に応じて、下位のストレージデバイス上に、大容量の VxFS ファイルシステムを格納できる十分な容量があることを確認し（詳しくは vxassist（1M）のマニュアルページを参照）、fsadm コマンドを使って VxFS ファイルシステムのサイズを変更します。

2. qiomkfile コマンドを使って、Quick I/O ファイルを拡張します。

```
$ /opt/VRTS/bin/qiomkfile -e extend_amount /mount_point/filename
```

または

```
$ /opt/VRTS/bin/qiomkfile -r newsize /mount_point/filename
```

次は、VxFS ファイルシステムを拡張する方法を示す例です。

/db01 を 500 MB にして、emp.dbf Quick I/O ファイルを 20 MB 拡張します。

```
$ /opt/VRTS/bin/qiomkfile -e 20M /db01/emp.dbf

# /opt/VRTS/bin/fsadm -b 500M /db01
```

次は、VxFS ファイルシステムを拡張する方法を示す例です。

/db01 を 500 MB にして、emp.dbf Quick I/O ファイルのサイズを 300 MB に変更します。

```
$ /opt/VRTS/bin/qiomkfile -r 300M /db01/emp.dbf

# /opt/VRTS/bin/fsadm -b 500M /db01
```

Quick I/O ファイルでの Oracle AUTOEXTEND の使用

Oracle では、指定した最大サイズに達するまで指定量ずつデータベースファイルを自動的に拡張する自動拡張機能がサポートされています。

通常のファイルシステムファイルの場合、下位のファイルシステムに十分な容量があれば、AUTOEXTEND は透過的に機能します。たとえば、データベースファイル emp.dbf のサイズが現在 100 MB で、時間の経過とともに 3 倍になると仮定します。AUTOEXTEND 機能を使ってこの拡張に対応する場合は、next サイズを 20 MB と指定し、maxsize を 300 MB と指定できます。これにより、ファイルのサイズが 300 MB になるまで、ファイルを 20 MB ずつ自動的に拡張できます。次に例を示します。

```
alter database datafile 'emp.dbf' autoextend on next 20m ¥
maxsize 300m;
```

（alter database コマンドと next および maxsize パラメータについて詳しくは『Oracle Server SQL リファレンスガイド』を参照してください。）

メモ: ファイルに対して AUTOEXTEND を実行するには、下位のファイルシステムに十分な容量が必要です。また、下位のストレージデバイスには、より大きい新しいファイルシステムを格納できる十分な容量が必要です。

Quick I/O ファイルまたは RAW デバイスの場合、AUTOEXTEND から下位の Quick I/O ファイルまたはデバイスを拡張することはできません。そのため、AUTOEXTEND でデータファイルを拡張するには、Quick I/O ファイルに、新しいサイズを格納できる十分なサイズが必要です。

Quick I/O ファイルでは、次のように AUTOEXTEND を使います。

- Quick I/O ファイルに、このデータベースファイルが拡張した結果として予測される最大サイズ以上を事前に割り当てます。
  この方法では、Quick I/O ファイル emp.dbf に 300 MB すべてを事前に割り当てる必要があります。そのため、余分なディスク容量が不要に確保されることがあります。RAW デバイスにも、同様の必要条件があります。
- Quick I/O ファイル内の使用可能な空き領域を監視し、必要に応じて qiomkfile コマンドでファイルを拡張します。
  RAW デバイスとは異なり、Quick I/O ファイルは簡単にオンラインで拡張できます。この方法では、Oracle データファイル内の使用可能な空き領域を監視し、必要に応じて（一般に、ファイルの使用率が約 80 から 90 % になった場合）qiomkfile コマンドによってオンラインで Quick I/O ファイルを拡張できます。この方法では、Quick I/O ファイル用に未使用のディスク容量を確保する必要がありません。ファイルシステム上の空き領域は、他のアプリケーションで使えます。

qiomkfile コマンドには次のオプションがあります。

-e	Oracle 表領域のサイズ変更を可能にするため、ファイルを指定量だけ拡張します。
-r	Oracle 表領域のサイズ変更を可能にするため、ファイルを指定サイズに拡張します。

fsadm コマンドを使うと、下位の VxFS ファイルシステムをオンラインで拡張できます（さらに下位のディスクまたはボリュームが拡張可能な場合）。詳しくは fsadm_vxfs（1M）のマニュアルページを参照してください。

Oracle 表領域内の使用可能な空き領域を監視するには

◆ 次の SQL 文を使って、Oracle 表領域内の使用可能な空き領域を確認します。

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> select * from dba_free_space where ¥
tablespace_name = 'tablespace_name';
SQL> exit
```

qiomkfile を使って Quick I/O ファイルを拡張するには

◆ データファイル内の空きブロックの割合が低い場合、qiomkfile コマンドを使って Quick I/O ファイルを拡張します。

```
$ /opt/VRTS/bin/qiomkfile -e extend_amount ¥
  /mount_point/filename
```

次の例で、ファイルシステム /db01 の表領域 EMP 上の空き領域を監視する方法を示します。

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> select * from dba_free_space where tablespace_name = 'EMP';
SQL> exit
```

次の例で、qiomkfile コマンドを使って、Oracle データファイル emp.dbf を 20 MB（指定した next サイズ）だけ拡張する方法を示します。

```
$ /opt/VRTS/bin/qiomkfile -e 20M /db01/emp.dbf
```

データベースのリストア後の Quick I/O ファイルの再作成

Quick I/O を使っており、データベースのリストアが必要になった場合は、qio_recreate コマンドを使うと、データベースの完全なリカバリを実行した後で Quick I/O ファイルを自動的に再作成できます。qio_recreate コマンドは、mkqio.dat ファイルを使います。このファイルには、データベースによって使われる Quick I/O ファイルとファイルサイズのリストが含まれています。

データベースのリカバリについて詳しくは、ご使用のデータベースソフトウェアに付属するマニュアルを参照してください。

qio_recreate コマンドを使って Quick I/O を再作成する前に、次の条件が満たされていることを確認してください。

Oracle の前提条件

- Quick I/O ファイルの再作成を試みる前にデータベースをリカバリします。
- データベース管理者または root のどちらでログインしても、qio_recreate コマンドを実行できます。
- qio_recreate コマンドを実行するディレクトリに、既存の mkqio.dat ファイルが存在する必要があります。
- ORACLE_SID 環境変数を設定する必要があります。

p.124 の「Oracle ファイルの Quick I/O ファイルへの変換」を参照してください。

使用に関する注意事項

- qio_recreate コマンドは従来の Quick I/O ファイルのみをサポートします。
- 詳しくは、qio_recreate(1M) のマニュアルページを参照してください。

データベースのリカバリ後に **Quick I/O** ファイルを再作成するには

◆ Oracle DBA として、次のように qio_recreate コマンドを使います。

```
$ /opt/VRTSdbed/bin/qio_recreate -T ora
```

コマンドが成功した場合は、何も出力されません。

qio_recreate コマンドを実行すると、次の処理が行われます。

状態...	処理内容...
Quick I/O ファイルが存在しない	Quick I/O ファイルが再作成されます。
通常の VxFS ファイルから Quick I/O ファイルへのシンボリックリンクが見つからない	シンボリックリンクが再作成されます。
シンボリックリンクとそれに関連する Quick I/O ファイルが存在しない	シンボリックリンクと Quick I/O ファイルがともに再作成されます。
Quick I/O ファイルが存在せず、それとシンボリックリンクでリンクされている通常の VxFS ファイルが元の VxFS ファイルではない	Quick I/O ファイルが再作成されず、警告メッセージが表示されます。
Quick I/O ファイルのサイズが mkqio.dat ファイルにリストされているサイズよりも小さい	Quick I/O ファイルが再作成されず、警告メッセージが表示されます。

Quick I/O の無効化

Quick I/O 機能を無効にする必要がある場合は、最初にすべての Quick I/O ファイルを通常の VxFS ファイルに戻す必要があります。次に、特別なマウントオプションを使って、VxFS ファイルシステムを再マウントします。

Quick I/O を無効にする前に、次の条件が満たされていることを確認してください。

前提条件

AIX の場合:

再マウントするファイルシステムは、`/etc/filesystems` ファイルに配置しておく必要があります。

Solaris の場合:

再マウントするファイルシステムは、`/etc/vfstab` ファイルに配置しておく必要があります。

Quick I/O を無効にするには

1. データベースが実行中の場合は、停止します。
2. Quick I/O ファイルを通常の VxFS ファイルに戻すには、`mkqio.dat` リストを含むディレクトリで、次のコマンドを実行します。

   ```
   $ /opt/VRTSdbed/bin/qio_convertdbfiles -u
   ```

   `mkqio.dat` ファイル内の Quick I/O ファイルのリストが表示されます。次に例を示します。

   ```
   .file1::cdev:vxfs: --> file1
   .file2::cdev:vxfs: --> file2
   .file3::cdev:vxfs: --> file3
   .file4::cdev:vxfs: --> file4
   .file5::cdev:vxfs: --> file5
   ```

   `qio_convertdbfiles` コマンドに、変換を元に戻すオプション (`-u`) を指定すると、ファイルの名前は .filename から filename に変更され、Quick I/O ファイルとともに作成された .filename へのシンボリックリンクは削除されます。
3. Quick I/O を無効にしてファイルシステムを再マウントするには、次のように、`mount -o noqio` コマンドを使います。

   AIX の場合:

   ```
   # /opt/VRTS/bin/mount -V vxfs -o remount,noqio special/mount_point
   ```

   Solaris の場合:

   ```
   # /opt/VRTS/bin/mount -F vxfs -o remount,noqio /mount_point
   ```

Solaris ローカルゾーンへの Quick I/O ファイルの作成

Quick I/O ファイルを Solaris ローカルゾーンに作成することはできません。これは、VxFS コマンド `qiomkfile` を Solaris ローカルゾーンで実行できないためです。

Quick I/O ファイルを Solaris ローカルゾーンに作成するには、次の回避策を実行する必要があります。次のコマンドは、root（特に明記されていない場合）として実行する必要があります。

Quick I/O ファイルを Solaris ローカルゾーンに作成するには

1. `VRTSodm` を Solaris グローバルゾーンにインストールします。

   この手順はシステムで 1 回のみ実行する必要があります。各ローカルゾーンでこの手順を繰り返す必要はありません。

2. Veritas InfoScale Storage Foundation 製品ライセンスを Solaris グローバルゾーンにインストールします。

   この手順はシステムで 1 回のみ実行する必要があります。各ローカルゾーンでこの手順を繰り返す必要はありません。

3. Quick I/O を使う予定の各 Solaris ローカルゾーンに、Solaris グローバルゾーンから Veritas InfoScale ライセンスをエクスポートします。

   これを行うには、次の一連のコマンドを入力します。

   ```
   zonecfg -z <zone_name>
   ```

   たとえば、Solaris ローカルゾーン名が zone2 の場合は、次のように入力します。

   ```
   # zonecfg -z zone2
     zonecfg:zone2> add fs
     zonecfg:zone2:fs> set dir=/etc/vx/licenses/lic
     zonecfg:zone2:fs> set special=/etc/vx/licenses/lic
     zonecfg:zone2:fs> set type=lofs
     zonecfg:zone2:fs> end
     zonecfg:zone2> verify
     zonecfg:zone2> commit
     zonecfg:zone2> exit
   ```

4 各ローカルゾーンで、odm デバイスを /dev/odm にマウントします。

これを行うには、Solaris グローバルゾーンで次のコマンドを使います。

```
# mkdir /zones/<zone_name>/dev/odm
```

次に、Solaris ローカルゾーンで次のコマンドを使います。

```
# mount -F odm /dev/odm /dev/odm
```

たとえば、Solaris ローカルゾーン名が zone2 の場合は、次のコマンドを使う必要があります。

```
# mkdir /zones/zone2/dev/odm
# mount -F odm /dev/odm /dev/odm
```

5 Oracle ユーザーとして、/opt/VRTS/bin/odmmkfile を使って Quick I/O ファイルを Solaris ローカルゾーンに作成します。ファイルの名前を Quick I/O ファイルに変更します。

これを行うには、次の一連のコマンドを入力します。

```
# su - <oracle_dba>
# cd <target_dir>
# /opt/VRTS/bin/odmmkfile -s <size> filename
# mv filename .filename
# ln -s .filename::cdev:vxfs: filename
# chown <Oracle_dba> .filename
```

13

Cached Quick I/O によるデータベースパフォーマンスの向上

この章では以下の項目について説明しています。

- Cached Quick I/O について
- Cached Quick I/O 設定のタスク
- ファイルシステム上での Cached Quick I/O の有効化
- Cached Quick I/O 対象ファイルの判断
- 個々のファイルに対する Cached Quick I/O の有効化と無効化

Cached Quick I/O について

Veritas Cached Quick I/O を使うと、大容量の未使用システムメモリを選択的なバッファリング機構で効率よく利用することによって、Veritas Quick I/O が持つデータベースパフォーマンスのメリットを維持および拡張できます。Cached Quick I/O では、ファイルシステムの先読みなど、バッファリング動作をサポートする機能もサポートされています。

適切な Quick I/O ファイルに対して Cached Quick I/O を有効に設定すると、データの格納にファイルシステムのバッファキャッシュが使われ、データベースのパフォーマンスが向上します。このデータストレージは、情報の検索時にシステムバッファキャッシュにアクセスし、ディスク I/O を回避することにより、システムの読み取り速度を向上させます。

データをキャッシュレベルで保持していると、次のようにデータベースパフォーマンスが向上します。

- 読み取り操作の場合、Cached Quick I/O にはシステムバッファキャッシュ内のデータベースブロックがキャッシュに保存されます。これにより、物理 I/O 操作数が低減できるため、読み取りのパフォーマンスが向上します。
- 書き込み操作の場合、Cached Quick I/O は、「直接書き込みおよび遅延コピー技術」を使って、データのバッファキャッシュコピーを更新しています。ダイレクト I/O のスケジュールが設定されてから I/O の完了を待機している間に、書き込まれる変更データが反映されるようにファイルシステムがバッファを更新します。オンライントランザクション処理の場合、Cached Quick I/O は、大容量の物理メモリを搭載した大規模プラットフォーム上での RAW デバイスよりも、データベースのスループットにおいて優れたパフォーマンスを実現します。
- 順次テーブルスキャンの場合、Cached Quick I/O は、Veritas File System で使われる先読みアルゴリズムにより、クエリー応答時間を大幅に短縮できます。ユーザーがファイル内の同じ領域を読み込む場合、データがキャッシュ内に残っていると、システムはディスク上のデータのスキャンを行わないで、キャッシュ内でヒットしたデータをすぐに返すことができます。

メモ: Cached Quick I/O は Linux ではサポートされていません。

Cached Quick I/O 設定のタスク

Cached Quick I/O を設定して使うには、次の処理を次の順番どおりに行う必要があります。

- データベースに使われる下位のファイルシステム上で Cached Quick I/O を有効に設定します。
- ファイル I/O 統計を作成するために、システムを本番環境で稼動させます。
- ファイルを使用中の環境で、ファイル I/O 統計情報を取得します。
- Cached Quick I/O による効果が得られるファイルを判断するために、ファイル I/O 統計を分析します。
- キャッシングによる効果がないファイルの Cached Quick I/O を無効にします。

メモ: Cached Quick I/O は Linux ではサポートされていません。

ファイルシステム上での Cached Quick I/O の有効化

Cached Quick I/O では、正しく機能するため、下位のシステム拡張機能として稼動している Veritas Quick I/O に依存しています。次の手順に従い、Cached Quick I/O を有効に適用できるように、必ず正しいセットアップを行ってください。

前提条件

- vxtunefs コマンドを使って、Cached Quick I/O を有効または無効にするには、ファイルシステムの動作を変更する権限が必要です。デフォルトでは、vxtunefs コマンドを実行するにはスーパーユーザー（root）権限が必要ですが、それ以外のシステムユーザーには必要ありません。スーパーユーザー（root）は、次のようにして、データベース管理者にこのコマンドの使用許可を与える必要があります。

  AIX と Solaris の場合:

  ```
  # chown root:dba opt/VRTS/bin/vxtunefs

  # chmod 4550 /opt/VRTS/bin/vxtunefs
  ```

  dba グループに属するユーザーには、vxtunefs コマンドを実行する権限が付与されます。強力なコマンドへのアクセス権を与える場合は、さらに厳密にユーザーを選択する安全性のより高い方法を採用することをお勧めします。
- ファイルシステム上で Quick I/O を有効に設定する必要があります。Quick I/O は、ファイルシステムのマウント時に自動的に有効に設定されます。

使用に関する注意事項

- Cached Quick I/O が有効な場合、データベースサーバーのスワップデバイスへのページングアクティビティを監視することをお勧めします。vmstat -I コマンドを使用して、スワップデバイスのページングを監視します。スワップデバイスのページングが見られたら、データベースパフォーマンスを向上させるために適切な AIX の VMM（Virtual Memory Manager）チューニングが必要です。

使用に関する注意事項

AIX の場合:

- Cached Quick I/O が有効な場合、データベースサーバーのスワップデバイスへのページングアクティビティを監視することをお勧めします。vmstat -I コマンドを使用して、スワップデバイスのページングを監視します。スワップデバイスのページングが見られたら、データベースパフォーマンスを向上させるために適切な AIX の VMM（Virtual Memory Manager）チューニングが必要です。

qio_cache_enable フラグの有効化および無効化

スーパーユーザー（root）として、ファイルシステムのマウント後に vxtunefs コマンドを使って、qio_cache_enable フラグを設定します。

ファイルシステムの qio_cache_enable フラグを有効にするには

◆ 次のように、vxtunefs コマンドを使います。

AIX と Solaris の場合:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=1 /mount_point
```

次に例を示します。

AIX と Solaris の場合:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=1 /db02
```

/db02 は、Quick I/O ファイルが格納されている VxFS ファイルシステムであり、qio_cache_enable フラグを 1 に設定することにより、Cached Quick I/O が有効になります。このコマンドを使うと、このファイルシステム上のすべての Quick I/O ファイルのキャッシュが有効になります。

ファイルシステムの qio_cache_enable フラグを無効にするには

◆ 次のように、vxtunefs コマンドを使います。

AIX と Solaris の場合:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=0 /mount_point
```

次に例を示します。

AIX と Solaris の場合:

```
# /opt/VRTS/bin/vxtunefs -s -o qio_cache_enable=0 /db02
```

/db02 は、Quick I/O ファイルが格納されている VxFS ファイルシステムであり、qio_cache_enable フラグを 0 に設定することにより、Cached Quick I/O が無効になります。このコマンドを使うと、このファイルシステム上のすべての Quick I/O ファイルのキャッシュが無効になります。

再起動およびマウントにわたる Cached Quick I/O 設定の永続化

/etc/vx/tunefstab ファイルにファイルシステムエントリを追加することにより、再起動およびマウント後も、Cached Quick I/O システム設定を残すことができます。

メモ: tunefstab ファイルは、ユーザーによって作成されたファイルです。ファイルの作成およびチューニングパラメータの追加について詳しくは、tunefstab(4)のマニュアルページを参照してください。

再起動後にファイルシステムを有効にするには

◆ ファイルシステムを /etc/vx/tunefstab ファイルに格納し、フラグエントリを設定します。

```
/dev/vx/dsk/dgname/volname qio_cache_enable=1
```

ここで、

- /dev/vx/dsk/dgname/volname は、ブロックデバイスの名前です。
- *dgname* は、ディスクグループの名前です。
- *volname* は、ボリュームの名前です。

次に例を示します。

```
/dev/vx/dsk/PRODdg/db01 qio_cache_enable=1
/dev/vx/dsk/PRODdg/db02 qio_cache_enable=1
```

/dev/vx/dsk/PRODdg/db01 は、ファイルシステムが常駐しているブロックデバイスです。

tunefstab(4)のマニュアルページに、チューニングパラメータの追加方法が説明されています。

tunefstab(4)マニュアルページを参照してください。

メモ: vxtunefs では、マウントポイントまたはブロックデバイスを指定します。tunefstab では、必ずブロックデバイスを指定します。

---

vxtunefs によるチューニング情報の取得

vxtunefs コマンドを使って、各ファイルシステムに対する qio_cache_enable フラグの設定を確認します。

qio_cache_enable フラグ設定に関する情報を取得するには

◆ grep コマンドと vxtunefs を組み合わせて使います。

AIX と Solaris の場合:

```
# /opt/VRTS/bin/vxtunefs /mount_point | grep qio_cache_enable
```

次に例を示します。

AIX と Solaris の場合:

```
# /opt/VRTS/bin/vxtunefs /db01 | grep qio_cache_enable
```

/db01 は、ファイルシステムの名前です。このコマンドにより、次のように qio_cache_enable 設定のみが表示されます。

```
qio_cache_enable = 0
```

また、vxtunefs コマンドを使って、I/O 特性やチューニング統計の詳細リストを取得することもできます。

vxtunefs(1) マニュアルページを参照してください。

すべての vxtunefs システムパラメータの情報を取得するには

◆ grep コマンドを使わず、vxtunefs コマンドを使います。

```
# /opt/VRTS/bin/vxtunefs /mount_point
```

次に例を示します。

```
# /opt/VRTS/bin/vxtunefs /db01
```

vxtunefs コマンドにより、次のような出力が表示されます。

```
 Filesystem i/o parameters for /db01
read_pref_io = 65536
read_nstream = 1
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 1
write_unit_io = 65536
pref_strength = 10
buf_breakup_size = 131072
discovered_direct_iosz = 262144
max_direct_iosz = 1048576
default_indir_size = 8192
qio_cache_enable = 0
```

```
odm_cache_enable = 0
write_throttle = 0
max_diskq = 1048576
initial_extent_size = 1
max_seqio_extent_size = 2048
max_buf_data_size = 8192
hsm_write_prealloc = 0
read_ahead = 1
inode_aging_size = 0
inode_aging_count = 0
fcl_maxalloc = 887660544
fcl_keeptime = 0
fcl_winterval = 3600
fcl_ointerval = 600
oltp_load = 0
delicache_enable = 1
thin_friendly_alloc = 0
dalloc_enable = 1
dalloc_limit = 90
```

vxtunefs(1)マニュアルページに、vxtunefs パラメータの詳細とチューニング手順が説明されています。

vxtunefs(1)マニュアルページを参照してください。

Cached Quick I/O 対象ファイルの判断

Cached Quick I/O による効果を得ることができるファイルを判断する方法はアプリケーションによって異なるため、アプリケーション別に評価する必要があります。したがって、Cached Quick I/O を使うのに適したファイルの候補を決定するために、次の手順を 2 回以上実行する必要がある場合もあります。

Quick I/O に適したファイルの候補を判断する前に、次の条件が満たされていることを確認してください。

前提条件
- ファイルシステムで Cached Quick I/O を有効にする必要があります。

使用に関する注意事項
- 詳しくは qiostat(1M)のマニュアルページを参照してください。

I/O 統計情報の収集

ファイルシステムで Cached Quick I/O を有効にしたら、統計情報を収集し、Cached Quick I/O のメリットを最大限に利用できるファイルを判断して指定する必要があります。

ファイルに Cached Quick I/O のメリットがあるかどうかの判別に必要な統計情報を収集するには

1. 次のコマンドを入力することにより、qiostat カウンタをリセットします。

   ```
   $ /opt/VRTS/bin/qiostat -r /mount_point/filenames
   ```

2. 通常の負荷の元でデータベースを 1 回の完全なサイクル（多くの場合、24 時間から48 時間）にわたって実行し、1 日の中の時間帯ごとに異なる用途（OLTP、レポート、バックアップなど）におけるシステム I/O のパターンとデータベーストラフィックを採集します。

3. データベースの実行中に qiostat -l を実行して、次のようにキャッシング統計を取得します。

   ```
   $ /opt/VRTS/bin/qiostat -l /mount_point/filenames
   ```

   または、-i オプションを使って、統計情報を指定間隔で取得します。

   ```
   $ /opt/VRTS/bin/qiostat -i n /mount_point/filenames
   ```

   ここで、n は秒単位です。

   次に例を示します。

   ファイルシステム /db01 上のすべてのデータベースファイルから I/O 統計情報を取得するには、次のコマンドを入力します。

   ```
   $ /opt/VRTS/bin/qiostat -l /db01/*.dbf
   ```

Oracle の I/O 統計情報について

qiostat コマンドの出力は、特定のファイル上で Cached Quick I/O を有効または無効にするかを決定する際に使う主要な情報源です。統計情報は、オブジェクトごとに 2 行ずつ出力されます。

情報の 2 行目は次のように定義されています。

- CREAD は、VxFS キャッシュからの読み取り回数（または、キャッシュアドバイザリが有効になっている Quick I/O ファイルの読み取り総数）です。
- PREAD は、キャッシュアドバイザリが有効になっている Quick I/O ファイルの読み取り回数です。
- HIT RATIO は割合（%）で表示され、CREAD の数から PREAD の数を引いて 100 を掛け、さらに CREAD の総数で割った値です。数式は次のようになります。

  ```
  (CREADs - PREADs) * 100/ CREADs
  ```

qiostat -l コマンドにより、次のような出力が表示されます。

```
                 OPERATIONS  FILE BLOCKS  AVG TIME(ms)
CACHE_STATISTICS

FILE NAME       READ  WRITE   READ WRITE    READ WRITE
CREAD  PREAD  HIT RATIO

/db01/cust.dbf   17128 9634   68509  38536   24.8  0.4
17124  15728  8.2

/db01/system.dbf   6     1      21    4      10.0  0.0
  6     6     0.0

/db01/stk.dbf    62552 38498  250213 153992  21.9  0.4
62567  49060  21.6

                 OPERATIONS  FILE BLOCKS  AVG TIME(ms)
CACHE_STATISTICS

FILE NAME       READ  WRITE   READ WRITE    READ WRITE
CREAD  PREAD  HIT RATIO

                 OPERATIONS  FILE BLOCKS  AVG TIME(ms)
CACHE_STATISTICS

FILE NAME       READ  WRITE   READ WRITE    READ WRITE
CREAD  PREAD  HIT RATIO
```

出力を分析して、キャッシュのヒット率が一定のしきい値を超える条件を検出します。特定のアプリケーションのファイルでキャッシュのヒット率が 20 % を超える場合は、そのファイルに対して Cached Quick I/O を使うことが有効である可能性があります。より負荷の大きいシステムの場合は、必要に応じてこの比率を 30 % 以上にします。キャッシュのヒット率のしきい値は、データベースの特性および負荷によって異なります。

前述のサンプル出力を例にすると、ファイル /db01/system.dbf では、キャッシュのヒット率が 0 のため、キャッシングが有効でないと判別されます。さらに、このファイルでは、サンプリング所要時間内でほとんど I/O が行われていません。

反面、ファイル /db01/stk.dbf では、21.6 % のキャッシュのヒット率が計測されています。システムと負荷を前提に考慮して、前述のキャッシュのヒット率が許容しきい値を超えると判断した場合は、データベースでキャッシングが有効であると考えられます。また、読み取りおよび書き込み操作に関して報告された数値も検討します。/db01/stk.dbf ファイルに対する読み取りおよび書き込み I/O 数を比較すると、読み取り数が書き込み数のほぼ 2 倍となっていることがわかります。Cached Quick I/O を使うと、書き込み操作より読み取り操作が多いファイルの場合に、パフォーマンスを最大まで高めることができます。

これら 2 つの要因を考慮すると、/db01/stk.dbf は、Cached Quick I/O を適用する有力な候補になります。

p.151 の「個々のファイルに対する Cached Quick I/O の有効化と無効化」を参照してください。

I/O 統計値上での先読みの影響

qiostat 出力内の CREAD 数は、Cached Quick I/O を含む呼び出されたすべての読み取り回数で、PREAD 数は、物理的な読み取り回数です。CREAD と PREAD（CREADS - PREADS）の違いは、ファイルシステムキャッシュ内のデータにヒットした読み取り数です。したがって、PREAD 数は常に、CREAD 数と同じかそれよりも少ないと考えられます。

ただし、PREAD カウンタは、ファイルシステムで先読みが実行された場合にも増加します。このような先読みは、ファイルシステムでシーケンシャルな読み取りが検出されると発生します。キャッシュにほとんどヒットしないほどサンプル時間が少ないような特殊な場合、qiostat による出力で、CREAD 数が PREAD 数より少なく表示される場合があります。この場合の CREAD および PREAD 値に対して計算されたキャッシュのヒット率は、Cached Quick I/O を有効化または無効化のどちらにするかの判断に使うと誤解を招く可能性があります。

このような場合は、複数のデータポイントセットを収集することで、複数の統計値に基づくより正確な判断を行うことができます。したがって、ファイルのうちの 1 つだけに高いキャッシュのヒット率が示された場合でも、指定された表領域内のデータファイルに対して Cached Quick I/O を有効にすることが効果的な場合があります。

その他の分析ツール

特定のファイルに対して Cached Quick I/O を有効にするかどうかを決定する際に使う主要な情報源は qiostat コマンドの出力ですが、qiostat と組み合わせて他のツールを使うこともお勧めします。たとえば、データベースのスループットを計測するソフトウェアのベンチマークテストなども便利です。特定のデータファイルセットに対して Cached Quick I/O を有効に設定した状態で、ベンチマークテストの結果がパフォーマンスの向上を示した場合、この結果に基づいて Cached Quick I/O を有効にすることも考えられます。

個々のファイルに対する Cached Quick I/O の有効化と無効化

qiostat またはその他の分析ツールを使って、Cached Quick I/O に適したファイルを決定した後、ファイルシステム内のキャッシングのメリットがないファイルに対して、qioadmin コマンドを使って、Cached Quick I/O を無効にする必要があります。

前提条件

- 特定のファイルレベルで Cached Quick I/O を有効または無効にする前に、そのファイルが存在するファイルシステムに対して Cached Quick I/O を有効に設定します。

使用に関する注意事項

- データベースがオンラインであっても、特定のファイルに対して Cached Quick I/O を有効または無効にできます。
- qiostat を使ってファイルを定期的に監視し、キャッシュのヒット率の変化がファイルに対する Cached Quick I/O の有効化または無効化を再検討するほどではないことを確認します。
- 同一ファイルシステム上の特定のファイルに対する Cached Quick I/O の有効化または無効化は、キャッシュアドバイザリを ON または OFF にして設定します。
- qioadmin(1)マニュアルページを参照してください。

個別のファイルに対するキャッシュアドバイザリの設定

個々のファイルに対するキャッシュアドバイザリの設定を変更することで、Cached Quick I/O をファイルごとに有効または無効にできます。

個々のファイルに対する Cached Quick I/O を無効にするには

◆ qioadmin コマンドを使って、次のようにキャッシュアドバイザリを OFF に設定します。

```
$ /opt/VRTS/bin/qioadmin -S filename=OFF /mount_point
```

たとえば、ファイル /db01/system.dbf に対して Cached Quick I/O を無効にするには、キャッシュアドバイザリを OFF に設定します。

```
$ /opt/VRTS/bin/qioadmin -S system.dbf=OFF /db01
```

個々のファイルに対する Cached Quick I/O を有効にするには

◆ qioadmin コマンドを使って、次のようにキャッシュアドバイザリを ON に設定します。

```
$ /opt/VRTS/bin/qioadmin -S filename=ON /mount_point
```

たとえば、qiostat を実行した結果、ファイル /db01/system.dbf のキャッシュのヒット率からキャッシングにメリットがあると判別されたとします。ファイル /db01/system.dbf に対して Cached Quick I/O を有効にするには、キャッシュアドバイザリを ON に設定します。

```
$ /opt/VRTS/bin/qioadmin -S system.dbf=ON /db01
```

個々のファイルの Cached Quick I/O 設定の永続化

/etc/vx/qioadmin ファイルにキャッシュアドバイザリエントリを追加すれば、再ブートや再マウントのときでも、個々のファイルに対する Cached Quick I/O 設定の有効化または無効化を永続させることができます。

qioadmin コマンドを使って設定されたキャッシュアドバイザリは、i ノード上に格納されるファイルの拡張属性として保存されます。この設定は、ファイルシステムの再マウントやシステムの再ブート後も持続しますが、この属性は、通常のバックアップ方法ではバックアップされないため、リストアできません。したがって、各ファイルリストア後は、常にキャッシュアドバイザリを再設定してください。Quick I/O ファイルに対するキャッシュアドバイザリを /etc/vx/qioadmin ファイルに保持している場合は、その必要はありません。

再ブートまたはマウント後に、個々のファイルの Cached Quick I/O 設定を自動的に有効または無効にするには

◆ 次のように、キャッシュアドバイザリエントリを /etc/vx/qioadmin ファイルに追加します。

```
 device=/dev/vx/dsk/<diskgroup>/<volume>

filename1,OFF

filename2,OFF

filename3,OFF

filename4,ON
```

たとえば、/db01 ファイルシステム内の個々のファイルに対する Cached Quick I/O 設定を永続化するには、/etc/vx/qioadmin ファイルを次のように編集します。

```
 #
 # List of files to cache in /db01 file system
 #
 device=/dev/vx/dsk/PRODdg/db01

 cust.dbf,OFF
 system.dbf,OFF
 stk.dbf,ON
```

qioadmin による個々のファイルの Cached Quick I/O 設定の判定

qioadmin コマンドを使ってファイルのキャッシュアドバイザリ設定を表示することにより、特定のファイルに対して Cached Quick I/O が有効または無効であるかどうかを確認できます。

メモ: Cached Quick I/O の設定を確認するには、各ファイルに対するキャッシュアドバイザリの設定と、vxtunefs を使ってフラグ qio_cache_enable の設定を確認する必要があります。

ファイルに対する現在のキャッシュアドバイザリ設定を表示するには

◆ 次のように、-P オプションを指定して qioadmin コマンドを使います。

```
$ /opt/VRTS/bin/qioadmin -P filename /mount_point
```

たとえば、/db01 ファイルシステム内のファイル cust.dbf に対する現在のキャッシュアドバイザリ設定を表示するには、次を入力します。

```
$ /opt/VRTS/bin/qioadmin -P cust.dbf /db01
```

```
cust.dbf,OFF
```

5

PITC の使用

- 第14章 PITC 方法の理解
- 第15章 Oracle PITC に関する注意事項
- 第16章 サードミラーブレークオフスナップショットの管理
- 第17章 領域最適化スナップショットの管理
- 第18章 Storage Checkpoint の管理
- 第19章 FileSnap スナップショットの管理
- 第20章 SFHA 環境での Netbackup によるバックアップとリストア

14

PITC 方法の理解

この章では以下の項目について説明しています。

- PITC（Point-In-Time Copy）の概要
- PITC を使う状況
- Storage Foundation PITC テクノロジについて
- SFDB ツールによってサポートされている PITC ソリューション
- Storage Foundation for Databases (SFDB) ツールのサポート対象スナップショットモードについて
- ボリュームレベルのスナップショット
- ボリュームレベルのスナップショット（FlashSnap）の逆再同期について
- Storage Checkpoint
- FileSnap について

PITC（Point-In-Time Copy）の概要

Storage Foundation は、業務上の重要なデータを管理するための柔軟で効率的な手段です。Storage Foundation を使うと、頻繁に更新されるデータベースのある瞬間におけるオンラインイメージ、すなわちポイントインタイムコピーを取得できます。

トランザクション処理、意思決定、知的財産の作成などに継続的に（週 7 日 24 時間）利用しなければならないデータ量がますます増えることが予想されています。消失や破壊からデータを保護することもますます重要になっています。以前は、データのバックアップが発生している間、データが変更されないように、データを停止状態にしていました。ただし、このオプションはダウンタイム最小化のニーズを満たしません。

PITC（ポイントインタイムコピー）はデータのオンライン可用性を最大化させます。PITC を使うと、システムのバックアップやアップグレード、その他の保守作業を行うことができま

す。PITC は、アクティブデータと同じホストまたは異なるホスト上で処理できます。必要な場合は、PITC の処理を別のホストにオフロードし、実稼動サーバーでのシステムリソースの競合を防止できます。この方法はオフホスト処理と呼ばれます。正しく実装されれば、オフホスト処理ソリューションが実稼動中のプライマリシステムの処理効率に影響を及ぼすことはほとんどありません。

PITC を使う状況

Veritas InfoScale FlashSnap による PITC ソリューションを用いるのに適した状況の例を次に挙げます。

- データバックアップ - 多くの企業では、365 日 24 時間データが使用可能でなくてはなりません。企業は、クリティカルなデータをオフラインでバックアップするときのダウンタイムを受け入れる余裕がありません。データのスナップショットを作成し、そのスナップショットからバックアップすれば、ダウンタイムを最小に抑え、処理効率に影響を与えることなく業務上の重要なアプリケーションを実行し続けることができます。
- データ継続性の提供 - プライマリストレージに障害が発生した場合に、サービスが継続するように、アプリケーションデータをリカバリするために PITC ソリューションを使用できます。サーバーにエラーが発生している場合、SFCFSHA または SFHA の高可用性クラスタ機能と共に、PITC（point-in-time copy）ソリューションを使うことができます。
- 意思決定支援システムの分析およびレポート作成 - 意思決定支援システムの分析や業務レポート作成などには、必ずしもリアルタイムな情報が必要というわけではありません。このような処理に対しては、プライマリデータベースへのアクセス競合が発生しないように、スナップショットから作成した複製データベースを使うよう設定できます。必要な場合は、複製データベースとプライマリデータベースを短時間で再同期することができます。
- テストとトレーニング - 開発グループまたはサービスグループは、スナップショットを新しいアプリケーションのテストデータとして使えます。開発者、テスト担当者、品質管理グループなどは、スナップショットデータを実際的な基準として、新しいアプリケーションの堅牢性、統合性および処理効率をテストできます。
- データベースエラーのリカバリ - 管理者やアプリケーションプログラムによって引き起こされる論理エラーによって、データベースの統合性が損なわれることがあります。Storage Checkpoint やスナップショットコピーを使ってデータベースファイルをリストアすると、テープなどのバックアップメディアからすべて修復するよりも短時間でデータベースを復旧させることができます。
  Storage Checkpoint を使って、データベースのインスタンスを過去のある時点の状態にすばやくロールバックすることができます。
- データのクローン - ファイルシステムまたはアプリケーションデータのクローンを作成できます。この機能を使用して、仮想デスクトップのプロビジョニングを迅速かつ効率的に行うことができます。

ここまでで説明したスナップショットソリューションはすべて、Volume Replicator と組み合わせてディザスタリカバリのサイトでも利用可能です。

レプリケーションを使用するスナップショットについて詳しくは、『Veritas InfoScale 7.4.2 レプリケーション管理者ガイド』を参照してください。

Storage Foundation は、次の使用例を含むいくつかの PITC ソリューションを、必要条件に応じて提供します。

- 意志決定支援システムの複製データベースの作成。
- スナップショットを使用したデータベースのバックアップとリカバリ。
- オフホストのクラスタファイルシステムのバックアップとリカバリ。
- オンラインデータベースのバックアップとリカバリ。

Storage Foundation PITC テクノロジについて

このトピックでは、Veritas FlashSnap™ 技術を活用した PITC（point-in-time copy）ソリューションの導入について説明します。Veritas FlashSnap テクノロジには Veritas InfoScale Enterprise ライセンスまたは Storage ライセンスが必要です。

Veritas InfoScale FlashSnap には、業務上の重要なデータを管理するためのフレキシブルで効率的な機能が用意されています。Veritas FlashSnap を使うと、変動するデータのオンラインイメージをある一時点で捉えて記録することができます。このイメージを「PITC」といいます。PITC を使うと、重要なデータの可用性を維持したまま、システムのバックアップ、アップグレードおよび他の保守作業を実行できます。必要な場合は、PITC の処理を別のホストにオフロードし、実稼動サーバーでのシステムリソースの競合を防止できます。

FlashSnap は、次の各種 PITC ソリューションをサポートしています。

- ボリュームレベルのソリューション。複数のタイプのボリュームレベルスナップショットがあります。これらの機能は、スナップショットを作成するときに別々のストレージが好ましいソリューションに適しています。たとえば、低層のストレージがあります。これらの技術のいくつかには、例外的なオフホスト機能が提供されています。
- ファイルシステムレベルのソリューションには、Veritas File System の Storage Checkpoint 機能が使われます。Storage Checkpoint は、次のようなストレージへのソリューションとして適しています。
  - ファイル数は少ないが、そのほとんどのファイルサイズが大きいという構成のファイルシステム
  - アプリケーションによるデータブロックの変更割合が少ないファイルシステム（たとえば、Web サーバーのコンテンツやデータベースなど）
  - アプリケーションのテストまたはバージョン管理のために書き込み可能なコピーを複数必要とするファイルシステム

p.170 の「Storage Checkpoint」を参照してください。

- ファイルレベルのスナップショット。
  FileSnap 機能は、個々のファイルレベルでスナップショットを提供します。

SFDB ツールによってサポートされている PITC ソリューション

Storage Foundation for Databases（SFDB）ツールは、Oracle データベースの PITC を作成するためにデータベース特有のコマンドラインを提供しています。SFDB ツールは Storage Foundation の下位機能を使用して、これらの操作を実行します。使いやすくするために、SFDB コマンドラインは少ない手順で Oracle データベースでの PITC 操作の実行を可能にしています。また SFDB コマンドラインでは、Oracle データベースに特有の機能を実行できます。

表 14-1 は、SFDB ツールによってサポートされている PITC ソリューションの比較を提供します。

表 14-1 PITC ソリューションの比較

	FlashSnap	領域最適化スナップショット	Database Storage Checkpoint	FileSnap
下位テクノロジ	ボリュームスナップショット（サードミラーブレークオフスナップショット）	ボリュームスナップショット（領域最適化インスタントスナップショット）	ファイルシステムチェックポイント	ファイルレベルのスナップショット（VxFS FileSnap）
オフホスト処理の可能性	はい	はい（Cluster Volume Manager を必要とする）	はい（Cluster File System を必要とする）	はい（Cluster File System を必要とする）
追加のストレージ要件	追加のミラープレックスが必要となります。プレックスは、元のデータのフルコピーです。	最小（コピーオンライトを使用し、キャッシュボリュームを必要とする）	最小（コピーオンライトを使用）	最小（共有エクステントを使用）
PITC 後のパフォーマンスインパクト	なし	コピーオンライトペナルティ	コピーオンライトペナルティ	書き込みによって分割が発生します。
単一の PITC からの複数のクローンのサポート	いいえ。ただし、複数のクローンを作成するのに異なるスナップショットによる異なるミラーを使用できます。	いいえ。ただし、複数のクローンを作成するのに複数のスナップショットを使用できます。	はい	はい
サポート対象のスナップショットモード	Online、Offline、Instant	Online、Offline、Instant	Online、Offline、Instant	Online

	FlashSnap	領域最適化スナップショット	Database Storage Checkpoint	FileSnap
Veritas Replicator によるレプリケーション環境のサポート	いいえ	はい（Instant スナップショットモードのみ）	いいえ	いいえ

Storage Foundation for Databases (SFDB) ツールのサポート対象スナップショットモードについて

SFDB ツールのサポートするスナップショットモードは次のとおりです。

- Online
- instant
- Offline

条件付きでサポートされる特定のスナップショットモードは、次のとおりです。

- FlashSnap、Storage Checkpoint、SOS は、すべてのモードをサポートします。
- FileSnap は、Online モードのみをサポートします。
- Online モードはすべてのスナップショット方法でデフォルトおよび推奨されるスナップショットモードです。

表 14-2 スナップショットモードの説明

スナップショットモード	説明
Online	オンラインスナップショットモード ■ アプリケーションまたはデータベースのオンラインまたはホットバックアップのようなもので、バックアップイメージとして適しています。 ■ アプリケーションでは、オンラインまたはホットバックアップモードをサポートする必要があります。Oracle データベースはスナップショットが取られるまでホットバックアップモードになります。 ■ データベースを開くように要求します。 ■ データベースは、アーカイブログモードである必要があります。 **メモ:** ほとんどの実稼働データベースは、アーカイブログモードです。 ■ アプリケーションまたはデータベースにパフォーマンスインパクトがほとんどありません。 ■ スナップショットイメージからのプライマリアプリケーションデータの復元を許可します。 ■ 復元操作後、ユーザーによるプライマリアプリケーションのスナップショットに基づいたクローンの手動による PIT 回復の実行を可能にします。
instant	インスタントスナップショットモード ■ アプリケーションまたはデータベースのクラッシュ一貫性イメージのようなもので、バックアップイメージとして適していません。 ■ クラッシュ一貫性スナップショットを作成できるように、アプリケーション I/O をフリーズします。アプリケーションファイルまたはデータを保持しているすべてのファイルシステムをフリーズすることによって、アプリケーション I/O がフリーズされます。アプリケーション I/O は、スナップショットプロセス中はフリーズした状態のままです。 ■ データベースは、アーカイブログモードである必要はありません。 ■ スナップショットが作成されている間にアプリケーション I/O はフリーズしているため、一部のアプリケーションまたはデータベースにパフォーマンスインパクトが生じます。ただし、データベースのトランザクション負荷が大きい場合は特に、スナップショットプロセス自体は早い可能性があります。 ■ スタンバイデータベースではサポートされていません。 ■ RAC データベースではサポートされています。

スナップショットモード	説明
Offline	オフラインスナップショットモード ■ アプリケーションまたはデータベースのコールドバックアップのようなもので、バックアップイメージとして適しています。 ■ アプリケーションがオフラインである必要があります。 ■ アプリケーションがオンラインのときスナップショット設定が検証されるように要求します。 ■ スナップショットモードでは、最速です。 ■ データベースは、アーカイブログモードである必要はありません。 ■ Oracle の場合、データベースインスタンスは実行されていないことが必要です。

ボリュームレベルのスナップショット

ボリュームスナップショットは、特定の時点での VxVM（Veritas Volume Manager）ボリュームのイメージです。ボリュームセットのスナップショットを作成することもできます。

スナップショットを使うと、ユーザーの作業中断を最小に抑えて、ボリュームのバックアップコピーをオンラインで作成できます。作成したバックアップコピーを使って、ディスク障害、ソフトウェア障害または人為的なミスが原因で失われたデータを復元したり、レポートの生成、アプリケーションの開発またはテストを行うための複製ボリュームを作成することができます。

ボリュームスナップショットを使って、オフホストオフラインバックアップも実装できます。

物理的には、データセットの完全なコピー（すべてのビットのコピー）の場合や、スナップショットが作成されてから更新されたデータセット要素だけが含まれる場合があります。後者の場合は、最初に書き込むときに割り当てる（allocate-on-first-write）スナップショットと呼ばれることもあります。もとのデータセット内の要素がはじめて更新（上書き）されるときにのみ、データ要素の領域がスナップショットイメージに追加されるためです。Storage Foundation の最初に書き込むときに割り当てる（allocate-on-first-write）スナップショットは、領域最適化スナップショットと呼ばれます。

ボリュームスナップショットの永続 FastResync

永続 FastResync がボリュームに対して有効にされると、VxVM は FastResync マップを使って、ボリュームやスナップショット内で更新されたブロックの履歴を残します。

スナップショットボリュームがもとのボリュームに再接続されると、永続 FastResync はスナップショットデータをただちに更新し再利用可能な状態にします。永続 FastResync はディスク領域上に FastResync マップを割り当てるため、システムやクラスタがクラッシュ

しても消失しません。専用ディスクグループ内のボリュームに対して永続 FastResync が有効にされると、ホストの再ブート後も更新分の再同期が実行されます。

永続 FastResync は、ボリュームとそのスナップショットボリュームの関係を、他のディスクグループへの移動後も追跡することができます。ディスクグループの再結合後は、永続 FastResync によってスナップショットプレックスを短時間で再同期できます。

ボリュームスナップショットのデータ整合性

ボリュームスナップショットは、特定時点でボリューム内に存在しているデータを取得します。したがって、上位のファイルシステムや、ファイルシステム内のファイルを開いているアプリケーション（データベースなど）によってメモリ上にキャッシュされているデータは、VxVM では認識されません。スナップショットの一貫性はクラッシュがあっても常に保持されます。つまり、アプリケーションにリカバリを実行させることで、スナップショットを使用可能にできます。これはサーバーのクラッシュ後にアプリケーションのリカバリが行われる方法と似ています。マウントされた Veritas File System（VxFS）を含むボリュームに対して、ボリュームの usetype 属性 `fsgen` が設定されている場合、VxVM は VxFS と連携して、キャッシュ内にあるデータをボリュームにフラッシュします。したがって、これらのスナップショットは常に VxFS に対して整合性があり、マウントするときに VxFS リカバリファイルが必要ありません。

データベースでは、適切な機構を追加的に使って、ボリュームスナップショットの作成時に表領域データの一貫性を確保する必要があります。最近のデータベースソフトウェアの多くは、ファイルシステムの I/O を一時的に停止する機能を備えています。この操作の実行方法は、このマニュアルに記載している例の中で説明しています。また、ファイルシステム内の通常のファイルは、さまざまなアプリケーションで開かれることがありますが、そのファイルデータの完全な整合性を確保するには、アプリケーションを停止し、ファイルシステムを一時的にマウント解除する以外に方法はありません。通常、整合性の確保が重要になるのは、スナップショットの作成時に使われていなかったファイルデータのみです。ただし、アプリケーションが調整されるすべての例では、スナップショットはクラッシュから回復可能です。

サードミラーブレークオフスナップショット

プレックスブレークオフスナップショットでは、スナップショットの作成に追加のミラーを使用します。1 つのプレックスボリュームにプレックスブレークオフスナップショットを作成できますが、通常はミラーボリュームのスナップショットを作成します。1 つのミラーボリュームには 1 つ以上のプレックス（ミラー）が存在し、各プレックスはデータのコピーです。スナップショット操作により、スナップショットボリュームになるプレックスの「ブレークオフ」、つまり切り離しが行われます。既存のプレックスを切り離したり、スナップショットミラーとして専用に機能する新しいプレックスを追加したりすることができます。通常は、元のボリュームに対して冗長性を維持します。元のボリュームが 2 つのプレックスを持つミラーボリュームの場合は、スナップショットのサードミラーを追加します。そのため、このタイプのスナップショップはサードミラースナップショットとも呼ばれます。

スナップショットプレックスは、同じディスクグループ内にあるボリュームの既存のプレックスとは異なるディスク上に存在する必要があります。ディスクには、既存のボリュームのコンテンツを含むのに十分なディスク領域がなければなりません。ボリュームが 1 TB の場合は、さらに 1 TB のディスク領域が必要になります。

スナップショットを作成すると、プレックスは 2 つのボリュームに分けられます。元のボリュームでは、元のプレックスが維持されます。スナップショットボリュームには、スナップショットプレックスが含まれます。元のボリュームでは、引き続き I/O を受け入れます。スナップショットボリュームでは、そのボリュームで処理を実行することを選択するまで、スナップショットが作成された時点でのデータを保持します。

スナップショットは複数作成できます。つまり、元のデータのコピーを複数持つことが可能です。

サードミラーブレークオフスナップショットは、領域最適化インスタントスナップショットやフルサイズインスタントスナップショットのコピーオンライト機構を使うと処理効率が低下する、書き込みを集中的に行うボリューム（データベース REDO ログなど）に適しています。

領域最適化インスタントスナップショット

領域最適化スナップショットには、元のデータオブジェクトが表現する完全な物理イメージは含まれないので、オフホスト代替処理用として選択することはできません。領域最適化インスタントスナップショットは、元のボリュームの変更部分のみをストレージキャッシュに記録します。元のボリュームに対する書き込みが発生すると、VxVM は書き込みをコミットする前にそのデータをキャッシュに保存します。ストレージキャッシュに必要なストレージのサイズは通常、元のボリュームよりもはるかに小さくて済むため、このスナップショットを領域最適化されていると呼びます。領域最適化スナップショットでは、スナップショットの存在期間中に元のボリューム上で更新されたデータ量に比例して、ストレージと I/O 帯域幅が消費されます。

領域最適化インスタントスナップショットの利点としては、即時使用や高速更新が可能であること、設定と管理が容易であることなどが挙げられます。完全コピースナップショットに比べて消費するストレージと I/O 帯域幅が少ないため、領域最適化スナップショットはより頻繁に作成できます。つまり、データ破損からのリカバリには適しています。

領域最適化スナップショットは、時間の経過とともに元のオブジェクトで変更されるデータが増えるため、自然に大きくなる傾向があります。したがって本質的に、有効期間が短い場合に適しています。

領域最適化スナップショットは、オフホスト代替処理用として選択することはできません。

領域最適化インスタントスナップショットの動作

領域最適化スナップショットでは、コピーオンライト機構を使って、最初の作成時またはデータの更新時にスナップショットをすぐに使用可能にします。

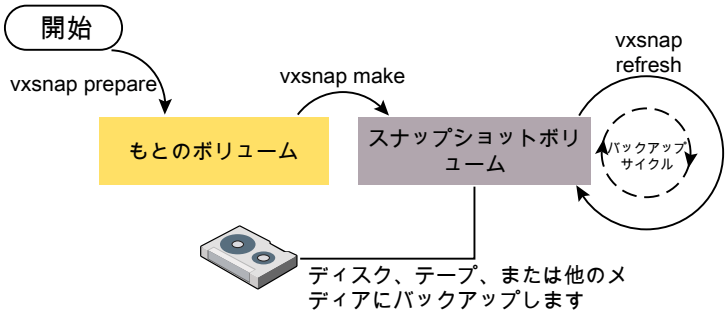
1 つのディスクグループに 1 つのストレージキャッシュを設定し、そのディスクグループ内のすべてのボリュームで共有できるようにすることが可能です。この場合、宣言するキャッシュの名前は、各ボリュームの領域最適化スナップショットで同一にする必要があります。キャッシュはディスク上に格納され、永続的です。

キャッシュがいっぱいに近づいた場合は、ディスクグループ内の任意の利用可能な空き領域を使ってキャッシュが自動的に拡大されるように VxVM を設定できます。

p.209 の「共有キャッシュオブジェクトの作成」を参照してください。

図 14-1 に領域最適化インスタントスナップショットのモデルを示します。

図 14-1　領域最適化インスタントスナップショットの作成とバックアップサイクルでの使用法



ボリュームレベルのスナップショット（FlashSnap）の逆再同期について

FlashSnap サービスでは、クローンを作成して、バックアップに使うことのできる Oracle データベースのスナップショットを作成できます。プライマリ Oracle データベースで何らかの破損が発生し、データベースが機能しなくなった場合、ボリュームスナップショットを使って、そのようなエラーから回復し、スナップショットの作成時の状態に復帰できます。FlashSnap サービスを使って作成したボリュームスナップショットからデータベースを回復する処理は逆再同期と呼ばれます。

Storage Foundation Database Flashsnap 機能を使って、Oracle 環境のデータベースの一時店のオンラインまたはオフライン PIT イメージを逆再同期できます。

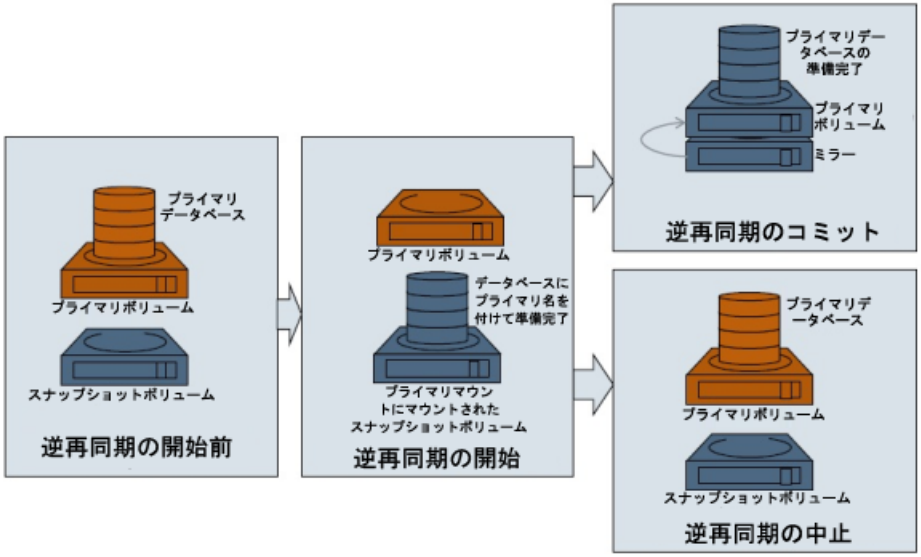
逆再同期とは、スナップショットミラーボリュームをプライマリデータボリュームにリストアすることを意味します。プライマリデータボリュームが論理的に破損したときに、スナップショットボリュームからの内容と置換する場合に逆再同期を使うことができます。

逆再同期操作は 3 つの特徴ある操作に分割されます。それらの 3 つの操作は次のとおりです。

- *逆再同期の開始*

- *逆再同期の中止*
- *逆再同期のコミット*

図 14-2 逆再同期の概要



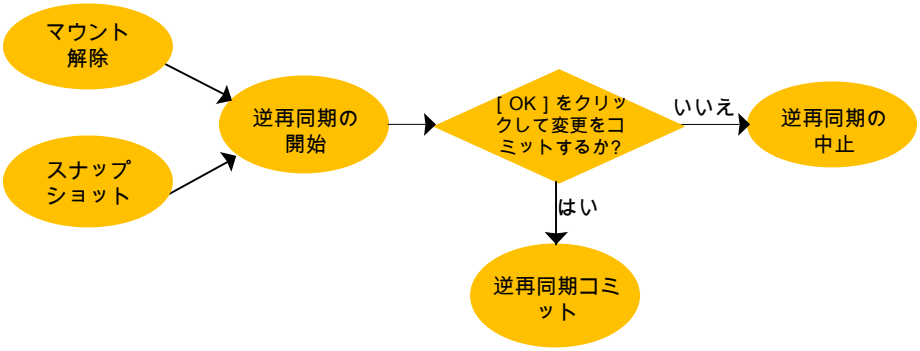
逆再同期操作のタイプ

逆再同期には、次の 3 つの操作があります。

- ReverseResyncBegin（エイリアス: RRBegin、RRStart、ReverseResyncStart）: *RRBegin* 操作を使ってスナップショットボリュームや Oracle インスタンスをマウントできます。
  スナップショットボリュームから Oracle インスタンスのプレビューを表示できます。
  ReverseResync Begin 操作は *manual*、*auto*、*auto_until_scn* の 3 つのリカバリ方法をサポートします。
- ReverseResyncAbort（エイリアス: RRAbort、RRCancel、ReverseResyncCancel）: この操作を使って、逆再同期プレビューセッションを終了し、元のデータボリュームに戻ることができます。
- ReverseResyncCommit（エイリアス: RRCommit、RRFinish、ReverseResyncFinish）: この操作を使って、*RRBegin* 操作でプレビュー表示した逆再同期の変更を実行できます。

メモ: この操作が完了すると、元のデータボリュームに戻ることはできません。スナップショットボリュームはデータボリュームに永続的にリスとあされます。逆再同期に使われる、スナップショットの後に作成されたストレージチェックポイントまたはファイルスナップは、RRCommit 操作後に利用できなくなります。

図 14-3 逆再同期操作とその動作



p.401 の 図 32-4 を参照してください。

逆再同期の制限事項

逆再同期には次の制限があります。

- 逆再同期は、オンラインまたはオフラインのスナップショットからのみ可能。インスタントスナップショットモードは逆再同期で使うことはできない
- RAC データベースではサポートされない
- スナップショットのクローンを作成した場合は行うことはできない
- アーカイブログボリュームがスナップショット操作から除外されている場合は、自動リカバリオプションは利用できない
- アーカイブログボリュームの逆再同期は、オフラインモードのスナップショットでは行うことができない
- 逆再同期はクラスタファイルシステム (CFS) 環境ではサポートされない
- 逆再同期は Data Guard 設定ではサポートされない
- 逆再同期は VVR (Veritas Volume Replicator) 設定ではサポートされない

逆再同期操作を実行するための前提条件

逆再同期操作を実行する前に次の事項を確認する必要があります。

HA 設定の逆再同期

高可用性（HA）環境で逆再同期を実行するには、データベースのサービスグルーをフリーズし、マウントリソースの VxFSMountLock 属性をゼロに設定する必要があります。

次に例を示します。

- `# hares -modify Mount_data VxFSMountLock 0`
- `# hares -modify Mount_arch VxFSMountLock 0`
- `# hagrp -freeze <DB_service_grp>`

逆再同期の完了後にデータベースのサービスグループをフリーズ解除し、元の値に属性をリセットできます。

逆再同期を実行する場合の推奨事項

- 逆再同期を実行する前に sfua_rept_util ツールを使ってリポジトリのバックアップを作成します。
- スナップショットを作成した後でデータベースの構造を変更する場合は（新しいデータファイルの追加など）、FlashSnap スナップショットコマンドを使ってバックアップを作成します。
- データベースのリカバリに RMAN バックアップを使う予定の場合は、データベースインカネーションの数を検討します。データベースインカネーション数は RRcommit 操作後に変わります。

5.X から逆再同期の実装に導入された変更

次に、逆再同期の実装に導入された変更を示します。

- ReverseResyncAbort 操作の後で ReverseResyncBegin 操作を実行できます。
- 新しいパラメータを使って ReverseResyncBegin 操作でデータベースのリカバリを次のように制御できます。

  `Reverse_Resync_Recovery`

  `Reverse_Resync_Archive_Log`

逆再同期操作のパラメータの設定

表 14-3 に逆再同期操作の新しいパラメータの一覧を示します。

表 **14-3** 逆再同期操作の新しいパラメータ

パラメータ	値	説明	コマンドラインオプション
REVERSE_RESYNC_RECOVERY	自動（Auto） 手動 Auto_Until_SCN	逆再同期時にデータベースをリカバリする方法を指定します。 デフォルトは AUTO です。自動データベースリカバリを試行するには自動リカバリオプションを使います。 -reverse-resync-begin と -reverse-resync-commit の手順の実行時にデータベースをマウントするには、手動リカバリオプションを使います。 スナップショットを作成した時点までデータベースを自動的にリカバリするには、AUTO_UNTIL_SCN オプションを使います。	--reverse_resync_recovery <value> または --rr_recovery <value>
REVERSE_RESYNC_ARCHIVE_LOG	いいえ はい。	アーカイブログボリュームを逆再同期するかどうかを指定します。 デフォルト値は NO です。値を NO に設定しないと、アーカイブログボリュームは逆再同期操作から除外されます。 値を YES に設定すると、アーカイブログボリュームは対応するスナップショットボリュームからリストアされます。*	--reverse_resync_archive_log <value> または --rr_archive_log <value>

メモ: *Reverse_Resync_Archive_Log = Yes オプションを設定すると、アーカイブしたログが壊れていて、バックアップからリストアする必要がある場合に役立ちます。

メモ: *Reverse_Resync_Archive_Log = No オプションを設定すると、ロールフォワードリカバリにプライマリデータベースのアーカイブログを使ってデータベースのリカバリを実行できます。

Storage Checkpoint

Storage Checkpoint は、特定時点におけるファイルシステムの永続イメージです。Storage Checkpoint は、コピーオンライト技術を使って、前回の Storage Checkpoint の作成後に変更されたファイルシステムブロックのみを特定し記録することにより、I/O オーバーヘッドを抑えます。Storage Checkpoint の重要な機能は次のとおりです。

- Storage Checkpoint は、システムに再ブートやクラッシュが発生しても、消失することはありません。
- Storage Checkpoint が作成されたときに対象ファイルシステム上にユーザーデータが存在している場合、Storage Checkpoint はファイルシステムのメタデータおよびディレクトリ階層に加えユーザーデータも保存します。
- マウントされたファイルシステムの Storage Checkpoint 作成後も、Storage Checkpoint のイメージに影響を与えずに、ファイルシステムのファイルを作成、削除および更新できます。
- ファイルシステムスナップショットとは異なり、Storage Checkpoint は書き込み可能としてマウントすることができます。
- ディスク領域の使用量を最小限に抑えるため、Storage Checkpoint はファイルシステム内の空き領域を使います。

各種データベース向けの Storage Foundation で提供されている Storage Checkpoint および Storage Rollback 機能は、データベースの破壊、ファイルの消失、表領域の削除などの論理エラーからデータベースを高速にリカバリします。データベースの連続的な Storage Checkpoint をマウントしてエラーを検出し、問題が発生する前の Storage Checkpoint にデータベースをロールバックすることができます。

Storage Checkpoint とスナップショットの違い

Storage Checkpoint は、次の点で Veritas File System スナップショットと異なります。

- Storage Checkpoint 自体に書き込み操作を実行することができる。
- システムの再ブートやシステム障害後も永続的に存在する。
- ファイルシステムと同じ空き領域プールを共有する。

- 最新の Storage Checkpoint の作成後に変更したファイルブロックのみを認識して、他の Storage Checkpoint との関連付けを管理する。
- 最新の Storage Checkpoint のみがプライマリファイルシステムから更新を蓄積するため、複数の読み取り専用 Storage Checkpoint を使うことで、I/O 操作と必要な格納領域を抑える。
- Storage Checkpoint 作成時点の状態にファイルシステムを復元できる。

Storage Checkpoint は、各種のバックアップおよびレプリケーションソリューションにおいて有効に活用できます。Storage Checkpoint には最後の Storage Checkpoint の後に変更されたファイルシステムブロックを追跡するための機能があるため、変更されたデータのみを取得すれば済むようなバックアップおよびレプリケーションアプリケーションの適用を容易にしています。Storage Checkpoint は、バックアップおよびレプリケーションソリューションの使用頻度を増やすことにより、データの移動を最小限に抑え、より高度な可用性およびデータの一貫性を実現します。

Storage Checkpoint は、多くのファイルを処理する環境（数百万のファイルを扱うファイルサーバーなど）でパフォーマンスにほとんど影響を与えることなく利用できます。ファイルシステムは Storage Checkpoint の作成時にフリーズされたままではないため、Storage Checkpoint が使われている場合でも、アプリケーションはファイルシステムにアクセスできます。ただし、Storage Checkpoint の作成は、ファイルシステム内のファイル数によっては時間がかかる場合があります。

Storage Checkpoint の動作

Storage Checkpoint 機能は、マウントされたファイルシステム（プライマリファイルセット）をフリーズし、Storage Checkpoint を初期化してから、ファイルシステムをアンフリーズします。この場合、まずファイルシステムを静的な状態にしてから、すべてのデータをディスクに書き込みます。フリーズ処理により、ファイルシステムへの I/O 操作がすべて遮断されます。Storage Checkpoint は実際のデータを含まずに作成されます。すなわち、Storage Checkpoint は、データではなく、プライマリファイルセットのブロックマップを示します。次に実行されるアンフリーズプロセスで、ファイルシステムへの I/O 操作が再開されます。

単一のファイルシステムまたは複数のファイルシステムに対して、Storage Checkpoint を作成できます。複数のファイルシステムの Storage Checkpoint の場合は、ファイルシステムを同時にフリーズし、ファイルシステムすべてに対して Storage Checkpoint を作成した後、ファイルシステムをアンフリーズします。その結果、複数のファイルシステムの Storage Checkpoint は同じ作成タイムスタンプを持ちます。Storage Checkpoint 機能は、操作の進行中にシステムクラッシュが発生しない限り、複数のファイルシステムの Storage Checkpoint は、指定したすべてのファイルシステムに作成されるか、または、どのファイルシステムにも作成されないか、のどちらかであることを保証します。

メモ: システムクラッシュが発生すると、アプリケーションの呼び出し時に Storage Checkpoint をクリーンアップします。

プライマリファイルセットの Storage Checkpoint には、最初はプライマリファイルセット内の既存のデータブロックへのポインタだけが含まれ、独自のデータブロックは割り当てられていません。

図 14-4 は、ファイルシステム `/database` とその Storage Checkpoint を示しています。Storage Checkpoint は、作成時にはプライマリファイルセットと論理的に同一ですが、実際のデータブロックは含まれていません。

図 14-4 プライマリファイルセットと Storage Checkpoint

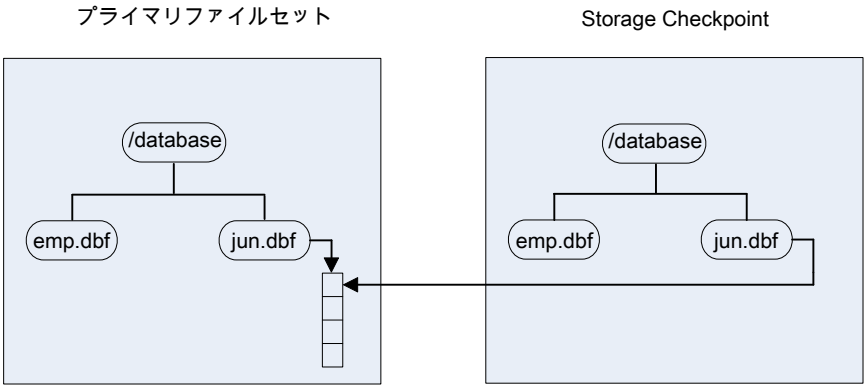
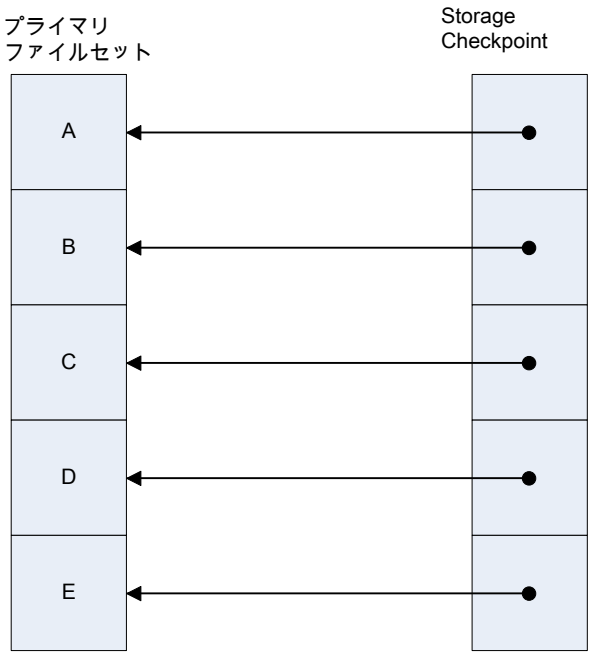


図 14-5で、四角はファイルシステムの各ブロックを表しています。この図の Storage Checkpoint には、図 14-4 で示されているように、Storage Checkpoint が作成された時点のプライマリファイルセットへのポインタが含まれています。

図 **14-5** Storage Checkpoint の初期化



Storage Checkpoint は、プライマリファイルセットからデータを検索することで、ファイルシステムと完全に一致するイメージを提供します。Storage Checkpoint は、VxFS のコピーオンライト技術を使って更新されます。

p.173 の「コピーオンライト」を参照してください。

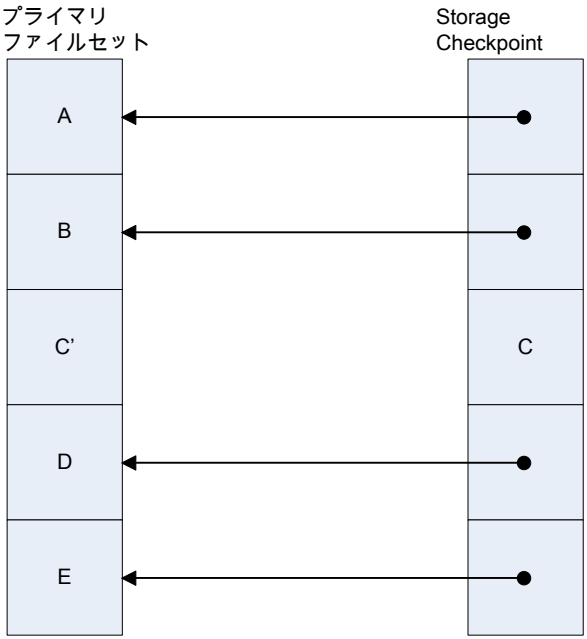
コピーオンライト

図 14-6 のプライマリファイルセットでは、最初は C が含まれていた 3 番目のデータブロックが更新されています。

新しいデータでデータブロックが更新される前に、もとのデータが Storage Checkpoint にコピーされます。この技術は、コピーオンライトと呼ばれるものです。これにより、Storage Checkpoint では Storage Checkpoint 作成時のプライマリファイルセットのイメージが維持されます。

更新や書き込み操作が行われるたびに、Storage Checkpoint にデータをコピーする処理が常に行われるわけではありません。古いデータを保存するのは 1 回だけでいいためです。プライマリファイルセットのブロックの更新が続くと、Storage Checkpoint にもとのデータブロックが蓄積されます。この例では、現在 C を含んでいる 3 番目のデータブロックがその後更新されても、Storage Checkpoint へのコピーは実行されません。これは、C を含むブロックのもとのイメージがすでに保存されているためです。

図 14-6　　プライマリファイルセットの更新



Storage Checkpoint の可視性

`ckptautomnt` マウントオプションによって、すべての Storage Checkpoint は、ディレクトリのリストに表示されない `.checkpoint` という特殊な名前を持つファイルシステムのルートディレクトリ内のディレクトリを介して、自動的にアクセス可能になります。このディレクトリの中には、ファイルシステムの各 Storage Checkpoint のディレクトリがあります。これらの各ディレクトリは、対応する Storage Checkpoint のマウントとして動作します。ただし、次の例外は除きます。

- NFS など外部アプリケーションは、ファイルをもとのマウントポイントの一部として参照します。したがって、追加の NFS エクスポートは不要です。
- アプリケーションに開示される i ノード番号は、マウントオプションに応じて一意にできます。

Storage Checkpoint は内部で自動マウントされますが、その自動マウントはオペレーティングシステムでは認識されません。これは、Storage Checkpoint を手動でマウントできず、マウントされたファイルシステムのリストに表示されないことを意味します。Storage Checkpoint が作成または削除されると、Storage Checkpoint ディレクトリのエントリは自動的に更新されます。Storage Checkpoint のファイルがまだ使用中の場合でも、Storage Checkpoint が `-f` オプションで削除されると、Storage Checkpoint は強制的にマウント解除され、ファイルでの操作はすべて EIO エラーで失敗します。

VxFS（Veritas File System）の古いバージョンで作成されたディレクトリが存在する場合や Storage Checkpoint への可視機能が削除された場合など、ファイルシステムのルートディレクトリに .checkpoint という名前のファイルやディレクトリがすでに存在する場合、Storage Checkpoint へのアクセスを提供する擬似ディレクトリにはアクセスできません。この機能が有効の場合、ルートディレクトリに .checkpoint という名前のファイルまたはディレクトリを作成しようとすると、EEXIST エラーで失敗します。

メモ: 自動マウントされた Storage Checkpoint が NFS マウントによって使われている場合、Storage Checkpoint の削除は、強制（-f）オプションを指定しなくても成功することがあります。

Storage Checkpoint と 64 ビットの i ノード番号

ファイルの i ノード番号は、Storage Checkpoint 全体で同じです。たとえば、ファイル file1 がファイルシステムにあり、Storage Checkpoint がそのファイルシステムを取る場合、元のファイルシステムと Storage Checkpoint の file1 で stat コマンドを実行すると、st_ino で同じ値が返されます。st_ino と st_dev の組み合わせは、システム内のすべてのファイルを一意に識別する必要があります。これは、Storage Checkpoint は別々にマウントされ、st_dev が異なるため、通常は問題ありません。Storage Checkpoint のファイルに Storage Checkpoint の可視性拡張子を介してアクセスする場合、st_dev は元のファイルシステムと同様に、すべての Storage Checkpoint で同一です。つまり、st_ino と st_dev を使用してもファイルを一意に識別できなくなったことを意味します。

通常は、システムのすべてのファイルを一意に識別する必要はありません。ただし、正しく機能するためには一意に識別する必要がなるアプリケーションもあります。たとえば、あるバックアップアプリケーションは、ファイルが別のファイルにハードリンクされているかどうか確認するために、両方のファイルで stat を呼び出し、st_ino と st_dev が同一であるかどうか調べる場合があります。Storage Checkpoint の可視性拡張子を介して 2 つのクローンを同時にバックアップするようにバックアップアプリケーションに指示があった場合、それらのファイルに含まれているデータが異なる場合でも、アプリケーションは誤って 2 つのファイルが同一であると推測します。

デフォルトでは、SF（Storage Foundation）は i ノード番号を一意にしません。ただし、一意の 64 ビットの i ノード番号の使用を有効にするために uniqueino マウントオプションを指定できます。このオプションは再マウント中には変更できません。

Storage Checkpoint を使用したデータベースのロールバックについて

各 Storage Checkpoint はファイルシステムの一貫したポイントインタイムイメージであり、また、このオンディスクバックアップに対するリストア機能を実装したアプリケーションが Storage Rollback です。Storage Rollback は、Storage Checkpoint に含まれる変更されたブロックをプライマリファイルシステムにロールバックすることによって、データベースのリストアを高速化します。

Checkpoint と Storage Rollback のプロセス

Storage Checkpoint は、現在マウントされているファイルシステム (プライマリファイルシステム) の「クローン」を作成する、ディスクおよび I/O 効率のよいスナップショット技術です。スナップショットファイルシステムと同様に、Storage Checkpoint は、Storage Checkpoint が作成された時点でのファイルシステムのスナップの正確なイメージとして表示されます。ただし、個別のディスク領域を使うスナップショットファイルシステムとは異なり、すべての Storage Checkpoint が、プライマリファイルシステムが存在する、同じ空き領域プールを共有します。

メモ: また、Database Storage Checkpoint を読み取り専用モードまたは、読み取りおよび書き込みモードとしてマウントすることによって、通常のファイルシステムと同様に Database Storage Checkpoint ファイルにアクセスできるようになります。

最初は、Storage Checkpoint にはデータが格納されていません。Storage Checkpoint には、i ノード一覧とプライマリファイルセットのブロックマップのみが格納されています。このブロックマップは、プライマリファイルシステムの実データを指します。Storage Checkpoint には i ノード一覧とブロックマップのみが必要で、データはコピーされないため数秒で作成でき、領域もほとんど必要ありません。

Storage Checkpoint 実行時は、読み取り要求をプライマリファイルシステム上のデータの検索、該当するブロックマップのバッファへの読み取り、データ要求プロセスへの返答をひとまとまりの処理として実行します。書き込み操作によって、プライマリファイルシステム上のデータブロックへの更新が要求されると、書き込み要求を処理する前に古いデータを Storage Checkpoint にコピーし、その後、プライマリファイルシステムが新しいデータで書き込み要求を実行します。このため、Storage Checkpoint には、Storage Checkpoint の作成時点のプライマリファイルシステムの正確なビューが保持されます。古いデータは、一度保存するだけで済むため、プライマリファイルシステム上のブロック n に対する書き込みがさらに行われても、Storage Checkpoint にコピーされることはありません。プライマリファイルシステム上のデータブロックが変更されると、Storage Checkpoint には、プライマリファイルシステムからコピーされた元のデータが順次書き込まれていきます。これにより、Storage Checkpoint のブロックマップのうちプライマリファイルシステム上の元のブロックを指すものが徐々に減少していきます。

Database Storage Rollback は、プライマリファイルシステム上のデータベース、表領域またはデータファイルを Storage Checkpoint で作成されたポイントインタイムイメージにリストアします。

Database Storage Rollback は、該当する Storage Checkpoint の「前の」イメージをプライマリファイルシステムにコピーすることで行われます。Storage Checkpoint と同様に、Database Storage Rollback では、ファイルレベルではなく、ブロックレベルでリストア処理が実行されます。Database Storage Rollback は `-o restore` 操作で `vxsfadm` コマンドを使って実行されます。

次に例を示します。

```
$ $ /opt/VRTSdbed/bin/vxsfadm -s checkpoint /
 -a oracle -o restore --checkpoint_name checkpoint1
```

マウントされた Storage Checkpoint は、次のようにさまざまなアプリケーションソリューションに使えます。

- バックアップ
- データ保全調査
- アップグレード
- データベースの修正
- データ複製ソリューション

Storage Checkpoint を読み取りおよび書き込みモードとしてマウントすると、このコマンドは、この Storage Checkpoint へのロールバックを行うことができないようにします。そのため、不正に修正された Storage Checkpoint データによって、データベースが破損することがありません。Storage Checkpoint を読み取りおよび書き込みモードでマウントする場合、元の Storage Checkpoint の「シャドウ」Storage Checkpoint が作成され、この「シャドウ」Storage Checkpoint が読み取りおよび書き込みモードでマウントされます。これにより、データベースを元の Storage Checkpoint にロールバックできるようになります。

Storage Checkpoint の種類

次の種類の Storage Checkpoint を作成することができます。

- 「Data Storage Checkpoint」
- 「Removable Storage Checkpoint」
- 「Non-mountable Storage Checkpoint」

Data Storage Checkpoint

Data Storage Checkpoint は、Storage Checkpoint の作成時のファイルシステムの完全なイメージです。この Storage Checkpoint には、ファイルシステムのメタデータおよびファイルデータブロックが含まれます。ファイルシステムの場合と同様に、Data Storage Checkpoint では、マウント、アクセスおよび書き込みを実行できます。Data Storage Checkpoint は、アクティブなファイルシステムの永続的で安定したイメージが必要なバックアップアプリケーションに有効です。Data Storage Checkpoint により、書き込み操作を実行するシステムやアプリケーションにオーバーヘッドが発生します。Data Storage Checkpoint の有効期間を制限することにより、システムリソースへの影響を最小限に抑えることができます。

Removable Storage Checkpoint

Removable Storage Checkpoint は、ファイルシステムの容量が不足する特定の状況下で、自動的に削除されるようにすることができます。

p.178 の「Storage Checkpoint の領域管理に関する注意事項」を参照してください。

`create` や `mkdir` などのユーザー操作中に、ファイルシステムの領域が不足すると、Storage Checkpoint がマウントされていても Removable Storage Checkpoint は削除されます。これにより、ディスクスペースの領域不足による中断なしに、アプリケーションを続行できます。Non-removable Storage Checkpoint は `ENOSPC` のような状況下では自動的に削除されません。Removable Storage Checkpoint のみを作成することをお勧めします。ただし、`fsadm` などの特定の管理操作中は、ファイルシステムの領域が不足しても Removable Storage Checkpoint は削除されません。

Storage Checkpoints はデフォルトで non-removable として作成されます。デフォルトの動作は、`vxtunefs -D ckpt_removable=1` コマンドを使用することで、VxFS が Removable Storage Checkpoint を作成するように変更できます。デフォルトで Removable Storage Checkpoint を作成するように設定した場合、Non-removable Storage Checkpoint は `fsckptadm -R create` *`ckpt_name mount_point`* コマンドを使用することで作成できます。

`vxtunefs`(1M) と `fsckptadm`(1M) のマニュアルページを参照してください。

Non-mountable Storage Checkpoint

Non-mountable Storage Checkpoint をセキュリティ機能として使用できます。これにより、他のアプリケーションによる Storage Checkpoint へのアクセスや変更を防ぐことができます。

Storage Checkpoint の領域管理に関する注意事項

Storage Checkpoint を含むファイルシステムで領域が不足すると、ファイルの削除や既存ファイルの上書きなど、一部の操作が失敗することがあります。システムで十分な領域を確保できない場合は、操作に失敗します。

通常、データベースではファイル用に格納領域が事前に割り当てられており、書き込み操作が失敗することはありません。`create` や `mkdir` などのユーザー操作中に、ファイルシステムの領域が不足すると、Removable Storage Checkpoint は削除されます。これにより、ディスクスペースの領域不足による中断なしに、アプリケーションを続行できます。Non-removable Storage Checkpoint は `ENOSPC` のような状況下では自動的に削除されません。Removable Storage Checkpoint のみを作成することをお勧めします。ただし、特定の管理操作（`fsadm` コマンドの使用、`qiomkfile` コマンドの使用、`fsckptadm` コマンドによる Storage Checkpoint の作成など）の実行中は、ファイルシステムの領域が不足しても Removable Storage Checkpoint は削除されません。

カーネルが Storage Checkpoint を自動的に削除する場合は、次のポリシーが適用されます。

- 削除する Storage Checkpoint 数をできる限り抑えようとします。
- Non-removable Storage Checkpoint を削除候補から除外します。
- Data Storage Checkpoint が無くなった場合に限り、Nodata Storage Checkpoint を選択します。
- 最も古い Storage Checkpoint から削除します。
- Storage Checkpoint を削除します（マウントされている場合でも）。このように削除された Storage Checkpoint では、新しい操作が失敗し、該当するエラーコードが表示されます。
- 最も古い Storage Checkpoint が Non-removable Storage Checkpoint の場合は、最も古い Removable Storage Checkpoint が削除用に選択されます。このような場合、データを Non-removable Storage Checkpoint に強制的に保存しなければならない可能性があり、これに失敗した場合は FULLFSCK としてファイルシステムがマーク付けされます。これを避けるには、Removable Storage Checkpoint のみを作成することをお勧めします。

FileSnap について

FileSnap は、同じファイルシステムに格納された、同じ名前空間内のファイルの原子的に領域が最適化されたコピーです。VxFS（Veritas File System）はファイルシステムのディスクレイアウトバージョン 8 以降でスナップショットをサポートします。

FileSnap は、ファイルシステムまたはボリュームより細かい詳細度のオブジェクトのスナップショットを作成する機能を備えています。ファイルシステムの名前空間の一部のスナップショットを作成する機能では、ファイルシステムに格納されるデータのアプリケーションベースまたはユーザーベースの管理が必要になります。これは、ファイルシステムが一連のユーザーまたはアプリケーションによって共有されるか、あるいはデータが同じファイルシステム内でさまざまな重要度に分類されるときに便利です。

すべての通常のファイル操作は FileSnap でサポートされ、VxFS は FileSnap を区別することはありません。

FileSnap のプロパティ

FileSnap は、管理者権限の必要なしに、所有するデータのスナップショットを作成する機能を非 root ユーザーに提供します。これにより、アプリケーションサイクルの適切な時点にスナップショットをスケジュール化することで、ユーザーやアプリケーションがデータのバージョン管理、バックアップ、および復元を行えるようになります。FileSnap からの復元は、スナップショットをソースファイルとして指定し、元のファイルを `vxfilesnap` コマンドの引数としての書き込み先ファイルとして指定することで簡単に行えます。

FileSnap の作成では、ソースファイルが読み取り専用としてロックされ、操作の間書き込み先ファイルが排他的にロックされて、スナップショットが原子的に作成されます。ファイルシステム内の残りのファイルは、FileSnap の作成が進行中に I/O を一時停止せずにアクセスできます。スナップショットの作成が進行中に、ソースファイルへの読み取りアクセスも中断されません。これにより、複数のユーザーやアプリケーションによる、強制的ではない方法によるファイルシステムの真の共有が可能になります。

ソースファイルと書き込み先ファイル間の名前空間の関係は、書き込み先のファイルパスを指定することでユーザーが実行する `vxfilesnap` コマンドによって定義されます。Veritas File System（VxFS）は、ソースファイルと書き込み先ファイルを区別することなく、これらの 2 つのファイル間の内部関係を維持することもありません。スナップショットが完了すると、ソースファイルと書き込み先ファイル間の唯一の共有プロパティは共有されるデータブロックとブロックマップになります。

ファイルの FileSnap の数は事実上無制限です。技術的な限度は、VxFS ファイルシステムによってサポートされるファイルの最大数、つまりファイルセットあたり 1,000,000,000 ファイルです。同じファイルから何千もの FileSnap が作成され、これらのスナップショットファイルのそれぞれが同時に何千ものスレッドによって読み取りおよび書き込みが行われた場合、FileSnap は上書きによって共有解除が発生した場合に共有ブロックの競合を防止する設計によって適切に拡大縮小されます。FileSnap の上書きによる共有ブロックの共有解除が発生した場合に見られるパフォーマンスは、従来のコピーオンライトのパフォーマンスより書き込みを割り当てる場合のパフォーマンスに近くなります。

ディスクレイアウトバージョン 8 以降では、ファイル間で共有するブロックまたはエクステントをサポートするために、参照カウントは各共有エクステントで追跡されます。VxFS は、遅延型のエクステントの共有および共有解除によって参照カウントの更新を処理します。また、いったん共有としてマーク付けされたエクステントは、すべての参照が消失するまで共有が解除されることはありません。これは FileSnap の作成パフォーマンスとデータエクステント共有解除のパフォーマンスを改善するためです。ただし、これにより、遅延再生の処理時点でのみ正確であるファイルシステムの共有ブロック統計が事実上生成されます。つまり、ファイルシステムおよびファイル上の共有エクステント統計は、ファイルシステムの状態によって無効である可能性があります。

FileSnap に対する同時 I/O

FileSnap の設計と実装により、同一のファイルの異なるスナップショットへの同時読み取りまたは書き込みが、あたかも独立したファイルであるかのように実行できるようになります。エクステントが同一のファイルのスナップショット間で共有されている場合でも、共有が同時 I/O に悪影響を及ぼすことはありません。

コピーオンライトと FileSnap

Veritas File System（VxFS）は、共有エクステントによって参照されるファイルの領域を上書きするときに【★遅延コピーオンライト★】を実行するオプションをサポートします。通常のコピーオンライトの実装には、古いデータの読み取り、新しいブロックの割り当て、古

いデータの新しいブロックへの同期コピーまたは書き込み、および新しいデータの新しいブロックへの書き込みが含まれます。この結果、最悪の場合 1 つ以上のトランザクションの割り当てが行われ、続いて読み取り、同期書き込み、および上書きのために必要な I/O 動作に適合する別の書き込みが行われる可能性.があります。このシーケンスにより、通常のコピーオンライト操作がコストのかかるものになります。VxFS の遅延コピーオンライトの実装の場合、新しく割り当てられたブロックに古いデータがコピーされず、したがって新しいデータがブロック全体をカバーする限り、古いデータを読み取る必要がありません。この動作と共有エクステントアカウント処理の遅延処理を組み合わせることにより、将来遅延コピーオンライトは書き込みの割り当て処理に匹敵する完全なものになります。ただし、サーバークラッシュの発生時に、サーバーが新しく割り当てられたブロックに新しいデータをフラッシュしなかった場合、上書きされた領域内のデータは、データをフラッシュする前にサーバーがクラッシュした際に書き込みを割り当てた場合のデータに似たものになる場合があります。これはデフォルトの動作ではありません。デフォルトの動作では、上書きされた領域内のデータは新しいデータまたは古いデータのいずれかになります。

FileSnap からの読み取り

通常の読み取り要求の場合、Veritas File System（VxFS）は、共有データブロックが任意の FileSnap またはソースファイルからアクセスできる場合であっても、特定の共有データブロックのページキャッシュ内のデータページの 1 つのコピーのみキャッシュに保存します。共有データページがキャッシュに保存されると、FileSnap またはソースファイルのいずれかによるその後の要求はページキャッシュから読み込まれます。これにより、ディスクへの重複読み取り要求が不要になり、アレイの I/O 負荷が軽減されます。また、ページキャッシュの重複が減少し、その結果何千もの FileSnap がアクセスされたときにキャッシュのチャーニングがほとんどなくなり、システムページキャッシュを効率的に使用できるようになります。

ブロックマップの断片化と FileSnap

ソースファイルのブロックマップはスナップショットファイルによって共有されます。以前に共有された領域でデータが上書きされると、書き込みが行われるファイルのブロックマップが変更されます。ソースファイルの共有データエクステントが同じ領域への上書き要求のサイズより大きい場合、書き込まれるファイルのブロックマップは一層断片化された状態になります。

バックアップと FileSnap

共有ブロックを含む VxFS ファイルシステムの完全バックアップでは、ソースファイルシステム内の物理ブロックへの論理参照の合計数と同じスペースがターゲットにも必要になる場合があります。たとえば、1000 個の FileSnap が作成された 20 GB ファイルを所有している場合、論理ブロック参照の合計数は約 20 TB になります。VxFS ファイルシステムは、ファイルとファイル内の 1000 個のスナップショットを格納するために 20 GB を少し超える物理ブロックのみ必要になる一方で、このファイルシステムは、バックアップターゲッ

トが重複排除をサポートしていないと仮定した場合、ファイルシステムのバックアップを作成するために、バックアップターゲット上に 20 TB を超える領域が必要になります。

15

Oracle PITC に関する注意事項

この章では以下の項目について説明しています。

- データベースレイアウトの注意事項
- サポート対象の Oracle 設定
- マルチテナントデータベースのサポートに関する考慮事項
- クローンデータベースの動作

データベースレイアウトの注意事項

次のデータベースレイアウトの注意事項は、Storage Foundation for Databases（SFDB）ツールを使用している場合に適用されます。

- すべてのデータベースファイルが VxFS ファイルシステムに存在する必要があります。データベースファイルには、制御ファイルのすべてのコピー、データファイル、一時ファイル、およびオンライン REDO ログファイルのすべてのコピーが含まれます。すべての下位ボリュームは、VxVM ボリュームである必要があります。
- 最低 1 つのアーカイブログの宛先が VxVM ボリュームの VxFS に存在する必要があります。
- オンラインのスナップショットモードでは、アーカイブログの宛先には、制御ファイル、データファイル、一時ファイル、またはオンライン REDO ログファイルが含まれていてはいけません。
- データベース PFILE または SPFILE の宛先は、指定されません。ご使用のデータベースが pfile を使っており、pfile が `$ORACLE_HOME/dbs` 以外の非デフォルトの場所に存在する場合、PFILE パラメータを使って明示的に pfile の場所を指定する必要があります。

- Oracle ホームディレクトリの宛先は、指定されません。
- OMF 命名規則を使用したデータベースは、全面的にサポートされています。
- Oracle ASM ストレージのデータベースは、サポートされていません。
- オペレーティングシステムの認証を使って Oracle データベースにアクセスするには、`vxsfadm` コマンドを有効にする必要があります（/ as sysdba）。
- `vxsfadm` 操作を実行したときに特定の操作が失敗することがあり、この障害からの回復を `vxsfadm` によって試行すると回復も失敗することがあります。これが原因で、設定がエラー状態になり、アプリケーション変更の一部は元に戻りません。このシナリオでは、エラーになった回復後に、手動で変更を元に戻す必要があります。
  たとえば、Oracle データベースのチェックポイントを作成するときは、データベースがバックアップモードになり、データベースがバックアップモードから削除された後でチェックポイントが作成されます。チェックポイント作成が失敗し、回復操作が失敗した場合は、データベースはバックアップモードのままになります。

サポート対象の Oracle 設定

サポート対象の Oracle バージョンについては、次のデータベースサポート表を参照してください。

https://www.veritas.com/bin/support/docRepoServlet?bookId=112632971-112632974-1&requestType=pdf

SFDB ツールは、次の制限を除く Oracle 単一インスタンスと Oracle RAC データベースを全面的にサポートしています。

- 物理スタンバイデータベースで Storage Foundation for Databases (SFDB) ツールを使用している場合、次の制限が適用されます。
  - Oracle RAC スタンバイデータベースは、サポート対象外です。
  - インスタントスナップショットは、サポート対象外です。オフラインとオンラインのスナップショットだけがスタンバイデータベースでサポートされます。
  - スタンバイデータベースは、読み取り専用モード (アクティブスタンバイ) で開かれているか、読み書きモードでアクティブにされている必要があります。

    メモ: この要件には、アクティブな Oracle Data Guard ライセンスが必要になることがあります。

  - スタンバイデータベースが読み取り専用モードで開かれている場合、物理スタンバイデータベースでスナップショットのクローンまたは作成などの操作の実行に SFDB ツールが使用される前に、REDO の適用サービスを一時的に停止する必要があります。

- 論理スタンバイデータベースは、サポート対象外です。スナップショットまたはクローン操作は、論理スタンバイとして機能するデータベースでは実行できません。
- Oracle ASM ストレージを使用した Oracle データベースは、サポート対象外です。
- Oracle RAC データベースは、Veritas Volume Replicator (VVR) セカンダリサイトで領域最適化スナップショットを作成するためにはサポートされていません。
  VVR による Oracle 障害回復設定は、サポートされています。SFDB ツールを使用して、セカンダリサイトで領域最適化スナップショットも作成できます。
- 表領域の追加または破棄やデータファイルの追加または破棄など、データベースへの構造上の変更が進行中である場合、SFDB ツールはポイントインタイムコピーの作成をサポートしません。ただし、ポイントインタイムコピーが作成済みであれば、データベースの状態にかかわらずいつでもクローンを作成できます。
- SFDB ツールはサポート対象のバージョンのマルチテナントデータベースでの `vxsfadm` 操作 (ポイントインタイムコピー) をサポートしています。SFDB ツールは個々のプラグ可能なデータベース (PDB) の操作をサポートしません。

マルチテナントデータベースのサポートに関する考慮事項

マルチテナントデータベースには次の考慮事項が適用されます。

- SFDB ツールは PDB 内のオフライン表領域をサポートしません。
- PITC 操作中に PDB が OPEN 状態でない場合は、逆再同期操作を実行できません。
- tbslist オプションで指定されたチェックポイントをロールバックすると、指定された tbsname を持つ CDB または PDB 内のすべての表領域がロールバックされます。
- PDB の PITC を取得できるのは OPEN 状態の場合みです。PDB のクローンを作成した場合、これらの PDB はクローン CDB で表示できますが、これらの PDB に関する操作は失敗します。
- どの PDB も OPEN 状態でない場合、インスタントスナップショットに関する CDB クローンはサポートされません。
- 一部の PDB が読み取り専用モードで開いている場合は、クローン操作に失敗します。Oracle Database バージョン 12.1.0.2 ではこの問題が観察されない可能性があります。
- 一部の表領域が読み取り専用モードの場合、クローン操作は失敗します。Oracle Database バージョン 12.1.0.2 ではこの問題が観察されない可能性があります。
- 逆再同期操作の実行後、クローン操作に失敗することがあります。

- いずれかの PDB が制限モードで開かれている場合、CDB で実行される操作に失敗することがあります。

クローンデータベースの動作

クローンデータベースでは、表領域とデータファイルに次の注意事項が適用されます。

- クローンデータベースでは、一部のデータファイルがオフラインでも、関連付けされた表スペースがオンラインの場合、データファイルは通常 RECOVER ステータスに表示されます。そのため、クローンデータベースでオンラインにするのに、さらに手動による回復がこのようなファイルに対して必要になる場合があります。
- 一部の表スペースまたはデータファイルが回復中の場合、スナップショットプロセスが失敗します。
- オフラインの表スペースは、クローンデータベース内でオフライン状態に保持されます。
- 読み取り専用の表スペースは、クローンデータベース内で読み取り専用状態に保持されます。
- オンラインの表スペースは、クローンデータベース内でオンライン状態に保持されます。

16

サードミラーブレークオフスナップショットの管理

この章では以下の項目について説明しています。

- クローン作成のための Database FlashSnap
- Database FlashSnap のホストとストレージの準備
- Database FlashSnap を使用したデータベースのクローンの作成
- ミラーボリュームのプライマリボリュームとの再同期
- クローンデータベースの手動リカバリ
- セカンダリホストでのデータベースのクローン作成

クローン作成のための Database FlashSnap

Veritas Database FlashSnap は、バックアップやオフホスト処理を行うための、データベースの PITC（Point-In-Time Copy）の作成に役立ちます。Database FlashSnap により、ボリュームのバックアップコピーをオンラインで作成でき、ユーザーの介入は最小限で済みます。

Database FlashSnap を使うと、頻繁に更新されるデータベースのある瞬間におけるオンラインイメージ、つまりスナップショットを取得できます。データベースのスナップショットコピーは、「データベーススナップショット」と呼ばれます。データベーススナップショットは、実稼動データベースと同じホスト上か、同じストレージを共有するセカンダリホスト上で使うことができます。データベーススナップショットは、バックアップ、データウェアハウス、決定サポートクエリーなどのオフホスト処理アプリケーションに使えます。スナップショットが必要なくなったときは、データベース管理者により、もとのスナップショットをプライマリホストに戻して、もとのデータベースボリュームとそのスナップショットを再同期処理させる

ことができます。Database FlashSnap コマンドは、コマンドラインインターフェースから実行します。

Database FlashSnap の利点

Database FlashSnap には、次の利点があります。

- データベーススナップショットは、実稼動データベースと同じホスト上か、同じストレージを共有するセカンダリホスト上で使うことができます。
- 多くの企業では、システム管理者の役割とデータベース管理者の役割ははっきりと分かれています。データベーススナップショットの作成には、一般的にスーパーユーザー（root）権限を必要としますが、データベース管理者は通常この権限を保有しません。Database FlashSnap ではスーパーユーザー権限を必要としないため、これは障害になりません。データベース管理者はデータベースの整合性のとれたスナップショットを簡単に作成できます。

Database FlashSnap のホストとストレージの準備

Database FlashSnap のホストとストレージを準備するために、次の情報を確認してください。

ホストの設定

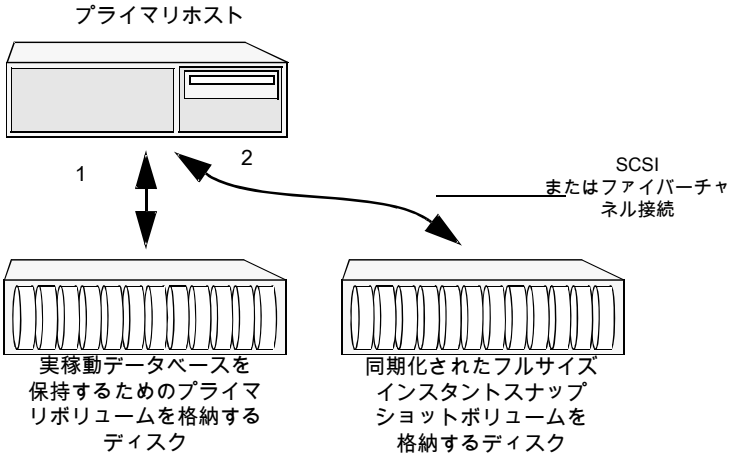
既存のデータベースと等しいサイズのミラーを追加するには、Database FlashSnap のディスクグループに十分なディスク領域が必要です。

Database FlashSnap 操作のためのストレージ設定を行うことはシステム管理者の責任であり、スーパーユーザー（root）権限が必要です。Database FlashSnap ユーティリティは適切なストレージ設定には対応していません。

同一ノード構成

図 16-1 に、プライマリホストに Database FlashSnap ソリューションを実装してディスクの競合を回避するための配置を提案します。

図 16-1 プライマリホストでの Database FlashSnap ソリューションの例

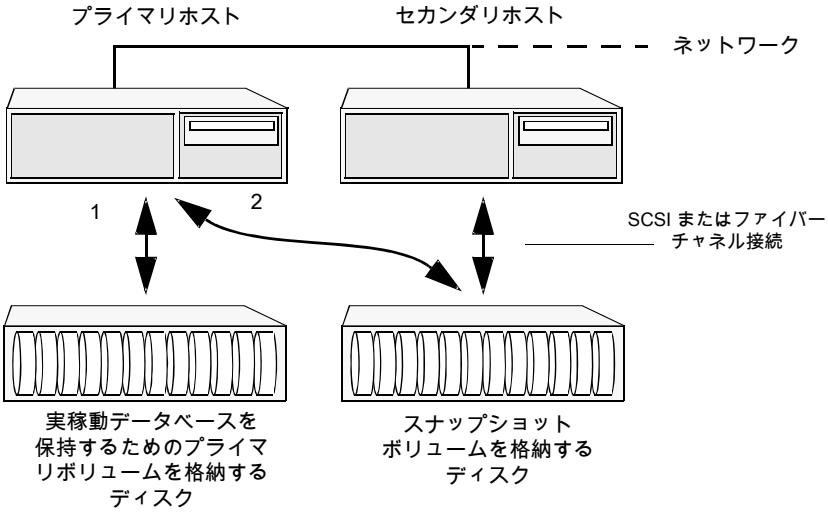


Database FlashSnap のオフホスト設定

Database FlashSnap のオフホスト設定では、実稼動データベースを実行しているプライマリホストのパフォーマンスを低下させることなく、オンラインバックアップと意思決定支援に対し CPU 集中型および I/O 集中型の操作を実行できます。プライマリホストとセカンダリホストの両方が、スナップショットデータベースが作成されるストレージを共有します。スナップショットボリュームを保持するディスクに、プライマリホストとセカンダリホストの両方からアクセスできます。

図 16-2 に、Database FlashSnap のオフホスト設定を示します。

図 16-2　　オフホスト Database FlashSnap ソリューションの例



メモ: VVR 環境で FlashSnap 機能を使うように計画した場合、VVR を設定した後で FlashSnap をセットアップするための前提条件の指示を実行してください。VVR の設定がすでに完了している場合は、ミラーまたはスナップショットボリュームのためのストレージが動作可能であることを確認してください。

オフホスト構成のホストおよびストレージの必要条件について詳しくは、次を参照してください。

p.54 の「データベースのオフホスト設定の必要条件」を参照してください。

データベースによって使われるボリュームまたはボリュームセットのスナップショットミラーの作成

Database FlashSnap を使うと、データベースで使っているボリュームを別のディスク群にミラー化できます。また、ミラー化されたものは、データベーススナップショットの作成に使えます。スナップショットボリュームは、個々のディスクグループに分割し、それぞれ配置できます。スナップショットディスクグループは、同じストレージがプライマリホストと共有している別のホストにインポートさせることができます。スナップショットボリュームは、プライマリボリュームを使って定期的に再同期を行い、データファイルの最新の変更分を取得できます。プライマリデータファイルが破損した場合は、スナップショットボリュームから破損したファイルを早急にリストアできます。スナップショットボリュームは、バックアップやリカバリ、クローンデータベースの作成など、さまざまな目的に使えます。

データベーススナップショットを作成する前にデータベースのデータファイルで使うボリュームすべてについてスナップショットミラーを作成する必要があります。この項では、ボリュームのスナップショットミラーを作成する手順を説明します。

スナップショットミラーの作成や同期には、vxsnap CLI コマンドを使います。

前提条件

- スーパーユーザー（root）としてログインする必要があります。
- ディスクグループは、バージョン 110 以上としてください。
  ディスクグループのバージョンについて詳しくは、vxdg（1M）のオンラインマニュアルページを参照してください。
- DCO（Data Change Object）と DCO ログボリュームが、スナップショットを作成するボリュームと関連付けられていることを確認してください。
- 永続 FastResync は、既存のデータベースボリューム上で有効にし、ディスクは、スナップショットボリュームに割り当てる必要があります。
  FastResync は、ミラーで見失った格納データに対する更新を追跡することにより、ミラーの再同期処理を最適にします。スナップショットミラーを、プライマリボリュームに再度取り付けた場合は、失われた更新分のみを再同期処理に再び適用させる必要があります。FastResync では、バックアップや意思決定支援などのサポート処理をさらに円滑にするために、ボリュームのスナップショット機構の処理速度をさらに向上させることができます。
  FastResync について詳しくは、『Storage Foundation 管理者ガイド』を参照してください。
- スナップショットミラーとそれに関連する DCO ログは、元のミラープレックスとは別のディスク上に配置し、システム管理者がスナップショットを作成できるように正しく設定を行う必要があります。
- スナップショットミラーを作成するときは、プライマリボリュームの別のコントローラやディスクにおいてスナップショットを作成します。
- アーカイブログに別のボリュームを割り当てます。
- $ORACLE_HOME/dbs ディレクトリに、制御ファイルを含むいかなるデータファイルも格納しないでください。

使用に関する注意事項

- Oracle データベースに関連するファイルについてそれぞれディスクグループを作成します。
- Oracle データベースファイルとその他のソフトウェアの間でボリュームを共有しないでください。
- ORACLE_HOME は、スナップショットミラーに格納できません。
- ブレークオフ時間中は、プライマリとスナップショットボリュームで変更されたデータ量に応じて再同期処理の速度が変化します。
- 元のミラーとスナップショットミラーとの間では、いかなるディスクも共有しないでください。
- データファイルとアーカイブログのスナップショットミラーは、それらが、元のボリュームのデータを含むいかなるディスクとも共有しないように作成される必要があります。このような方法で作成しなかった場合は、VxVM ディスクグループを分割できず、その結果、Database FlashSnap は機能しなくなります。

メモ: Database FlashSnap のコマンドでは、サードミラーブレークオフスナップショットのみをサポートしています。スナップショットミラーの状態は、SNAPDONE とする必要があります。

次の例に示す手順は、既存のスナップショットプレックスを持たない既存のボリューム、または関連するスナップショットボリュームを対象としています。この手順では、*volume_name* は、ボリュームまたはボリュームセットのいずれかの名称になります。

メモ: 次の手順でコマンドを実行するには、スーパーユーザー（root）としてログインする必要があります。

ボリュームまたはボリュームセットのスナップショットミラーを作成するには

1. スナップショットとするためのボリュームを準備するには、vxsnap prepare コマンドを使って次のように入力します。

```
# vxsnap -g diskgroup prepare volume ¥
alloc="storage_attribute ..."
```

vxsnap prepare コマンドでは、DCO と DCO ボリュームを自動作成し、それらをボリュームに対して関連付けを行い、そのボリュームの永続 FastResync を有効にします。永続 FastResync が有効になっているボリュームからスナップショットを生成した場合は、そのスナップショットでも永続 FastResync が自動的に設定されます。

ボリュームにおいて永続 FastResync を有効にするには、コマンドラインまたはスクリプト内から前述の vxsnap prepare コマンドを実行します。

2. ボリューム上の FastResync が有効となっていることを確認するには、vxprint コマンドを使って次のように入力します。

```
# vxprint -g diskgroup -F%fastresync volume_name
```

このコマンドの実行の結果、FastResync が有効だった場合には、on が返されます。それ以外の場合は off が返されます。

3. DCO と DCO ログボリュームが、ボリュームに関連付けられていることを確認するには、vxprint コマンドを使って次のように入力します。

```
# vxprint -g diskgroup -F%hasdcolog volume_name
```

このコマンドの実行により、DCO と DCO ログボリュームが、ボリュームに関連付けられている場合は、on を返します。それ以外の場合は off が返されます。

4. ボリュームのミラーを作成するには、次を実行します。

```
# vxsnap -g diskgroup addmir volume_name alloc=diskname
```

特定のボリュームのミラーを 3 つ作成する例

```
# vxsnap -g diskgroup addmir datavol ¥
nmirror=3 alloc=disk1,disk2,disk3
```

5. 使えるミラーを一覧で表示するには、次のように入力します。

```
# vxprint -g diskgroup -F%name -e"pl_v_name in ¥"volume_name¥""
```

6. Database FlashSnap により、スナップショットを作成したときに正しいミラープレックスが配置されるようにします。

- ミラーのブレークオフに使うデータプレックスに dbed_flashsnap タグを設定します。タグ名には任意のものを選択できますが、そのタグ名は、設定またはスナッププランで指定した SNAPSHOT_PLEX_TAG 属性と一致させる必要があります。

```
# vxedit -g diskgroup set putil2=dbed_flashsnap plex_name
```

- dbed_flashsnap タグが、必要なデータプレックスに設定されていることを確認するには、次のように入力します。

```
# vxprint -g diskgroup -F%name -e"pl_v_name in ¥
¥"volume_name¥" && p2 in ¥"dbed_flashsnap¥""
```

スナップショットにおいてデータのバックアップを必要とする場合は、適切なユーティリティまたはオペレーティングシステムのコマンドを使って、スナップショットの内容をテープまたはその他のバックアップメディアにコピーします。

Database FlashSnap を使用したデータベースのクローンの作成

「図 16-3」で概説されている手順を実行することにより、Database FlashSnap を使用してデータベースのクローンを作成できます。

図 16-3　クローン作成のワークフロー

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.329 の「FlashSnap の設定パラメータ」を参照してください。

p.332 の「FlashSnap がサポートされる操作」を参照してください。

詳しくは、vxsfadm-flashsnap(1M)と vxsfadm-oracle-flashsnap(1M)のマニュアルページを参照してください。

FlashSnap を使用して Oracle データベースのクローンを作成するには

1 設定ファイルを作成します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap ¥
-o setdefaults -a oracle --oracle_sid sfaedb ¥
--oracle_home /ora_base/db_home -c TEST1
Written config file TEST1
```

このコマンドは、-c オプションに指定された引数で示された場所にデフォルトの設定ファイルを作成します。パスが指定されていない場合は、現在の作業ディレクトリ内にすべてのパラメータとデフォルト値を含む設定ファイルが作成されます。必要に応じてパラメータを変更できます。

メモ: 指定された名前の設定ファイルがすでにある場合は、一意の設定名を指定するか、異なるパスを指定するか、既存の設定ファイルを削除します。削除操作を実行しても、この設定ファイルは削除されません。

---

2 設定を検証します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap ¥
-o validate -a oracle --oracle_sid sfaedb ¥
--oracle_home /ora_base/db_home --flashsnap_name ¥
TEST1 --app_mode online
Validating database configuration for third-mirror-break-off
snapshot:
ORACLE_SID = sfaedb
ORACLE_HOME = /ora_base/db_home
APP_MODE = online
SNAPSHOT_ARCHIVE_LOG = yes
REVERSE_RESYNC_ARCHIVE_LOG = no
REVERSE_RESYNC_RECOVERY = auto
ARCHIVELOG_DEST = /arch/sfaedb
Database validation successful.
Validating database volume layout for third-mirror-break-off
snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup datadg mounted on /data.
Archivelog volume ready for snapshot:
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
Storage units to be used for snapshot from diskgroup datadg:
vmdk0_25 vmdk0_27
SNAPSHOT_VOL_PREFIX = SNAP_
SNAPSHOT_DG_PREFIX = SNAP_
Database volume layout validated successfully.
```

このコマンドは設定ファイルとデータベース環境を検証します。問題が発生した場合、適切なエラーメッセージが表示されるため、このメッセージを使って問題を修正してから再試行できます。

3 データベースのスナップショットを作成します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap ¥
-o snap  -a oracle --oracle_sid sfaedb --oracle_home ¥
/ora_base/db_home --flashsnap_name TEST1 ¥
--app_mode online
Validating database configuration for third-mirror-break-off
snapshot:
ORACLE_SID = sfaedb
ORACLE_HOME = /ora_base/db_home
APP_MODE = online
ARCHIVELOG_DEST = /arch/sfaedb
SNAPSHOT_ARCHIVE_LOG = yes
REVERSE_RESYNC_ARCHIVE_LOG = no
REVERSE_RESYNC_RECOVERY = auto
Database validation successful.
snapshot started at Mon Aug 26 10:33:35 2013.
Putting database in backup mode ... Done
Validating database volume layout for third-mirror-break-off
snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup datadg mounted on /data.
Archivelog volume ready for snapshot:
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
Storage units to be used for snapshot from diskgroup datadg:
emc0_7 emc0_5
SNAPSHOT_VOL_PREFIX = SNAP_
SNAPSHOT_DG_PREFIX = SNAP_
Database volume layout validated successfully.
Creating snapshot volumes for data volumes ... Done
Taking database out of backup mode ... Done
Flushing online redo logs ... Done
Creating snapshot volume for archivelog volume ... Done
Copying snapshot information to snapshot volume ... Done
Creating snapshot diskgroups ... Done
Deporting snapshot diskgroups ... Done
SNAP_datadg
snapshot ended at Tue Aug 27 00:33:54 2013.
```

このコマンドはプライマリボリュームからのユーザー指定のミラー（パラメータ SNAPSHOT_PLEX_TAG）を破棄し、snap_dg_prefix パラメータで定義された文字列で開始される名前を持つ新しいディスクグループを作成します。このパラメータのデフォルト値は SNAP_ です。

スナップショットを作成する場合、切り離されたミラーはスナップショットディスクグループと言われる異なるディスクのグループに分割されます。ミラーはスナップショットディスクグループに存在する唯一のコンポーネントです。キャッシュオブジェクトのような追加のオブジェクトが、元のディスクグループからスナップショットディスクグループに移る必要がある場合、EXTRA_OBJECTS パラメータを使用してこれらの追加のオブジェクトを指定することができます。

p.329 の「FlashSnap の設定パラメータ」を参照してください。

メモ: 検証の段階で、必須のパラメータである `--oracle_sid` と `--flashsnap_name` を含め、すべてのパラメータが読み取られてリポジトリに格納されます。

パラメータを変更する必要がある場合は、設定ファイル内の変更を `-c` オプションで指定します。

4 スナップショットをマウントします。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap ¥
-o mount -a oracle --oracle_sid sfaedb ¥
--oracle_home /ora_base/db_home --flashsnap_name TEST1
Retrieving snapshot information ...                          Done
Importing snapshot diskgroups ...                            Done
Mounting snapshot volumes ...                                Done
```

メモ: このコマンドは、ストレージが追加されるホストにスナップショットをマウントします。セカンダリホストは、設定ファイルの SECONDARY_HOST パラメータで定義されるシステムです。

デフォルトでは、ボリュームは /var/tmp ファイルシステムの下にマウントされます。

スナップショットボリュームをマウントする場所として代替場所を指定する必要がある場合、コマンドラインまたは設定ファイルから CLONE_PATH を指定します。

オフホスト操作を実行するために、vxsfadm コマンドの -r オプションを使って SFDB リポジトリのホストを指定します。

```
$ vxsfadm -s flashsnap -a oracle -o mount ¥
--oracle_sid sfaedb --flashsnap_name TEST1 --clone_path
/cloneoracle
```

メモ: /cloneoracle ディレクトリが存在しない場合、Oracle ユーザーがこのディレクトリを作成するために必要な権限を持っていることを確認します。

---

5 スナップショットに基づいてデータベースのクローンを作成します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -o clone ¥
-a oracle --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1 --clone_name clone1
Retrieving snapshot information ... Done
Importing snapshot diskgroups ... Done
Mounting snapshot volumes ... Done
Preparing parameter file for clone database ... Done
Mounting clone database clone1 ... Done
Starting database recovery ...
Clone database clone1 is open.
```

clone_name を指定しなかった場合、自動的に生成されます。

メモ: 検証の操作中に使用した設定ファイルで *clone_name* パラメータおよび *clone_path* パラメータをすでに指定済みである場合は、コマンドラインに *clone_name* のオプションを含める必要はありません。

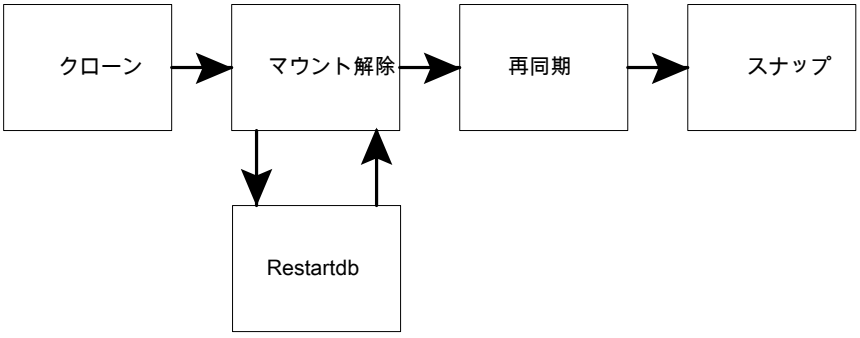
---

memory_target などのメモリチューニングパラメータを変更する必要がある場合、クローン操作中に変更済みの pfile を指定できます。

ミラーボリュームのプライマリボリュームとの再同期

データベースのクローンを作成した後で、図 16-4 で概説している手順を使ってミラーボリュームをプライマリボリュームで更新できます。

図 16-4 ミラーボリュームの再同期



ミラーボリュームをプライマリボリュームと再同期するには

1 クローンデータベースをマウント解除します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -o umount ¥
 -a oracle --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1 --clone_name clone1
Shutting down clone database clone1 ...
Database shut down.
Retrieving snapshot information ...                          Done
Unmounting snapshot volumes ...                              Done
Deporting snapshot diskgroups ...                            Done
```

このコマンドは、クローンデータベースを適切に停止してファイルシステムをマウント解除します。マウント解除したデータベースは、クローン操作を使って再起動できます。

`-o clone` オプションを使って、マウント解除した後にクローンデータベースを再起動できます。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -o clone ¥
-a oracle --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1 --clone_name clone1
Retrieving snapshot information ...                          Done
Importing snapshot diskgroups ...                            Done
Mounting snapshot volumes ...                                Done
Starting clone database clone1 ...
Clone database started.
```

このコマンドは、スナップショットファイルシステムをマウントしてクローンデータベースを再起動します。

2 ミラーボリュームをプライマリボリュームで更新します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -o resync ¥
-a oracle --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1
resync started at Sun Sep 18 14:53:17 2011.
Importing snapshot diskgroups ...                          Done
Joining snapshot diskgroups to original diskgroups ...     Done
Reattaching snapshot volumes to original volumes ...       Done
resync ended at Sun Sep 18 14:53:30 2011.
```

このコマンドは、スナップショット状態中に分割したすべてのミラーボリュームをプライマリボリュームと再同期します。再同期操作が完了してミラーボリュームが SNAPDONE 状態になれば、新規のスナップショットを取得できます。

`vxprint` コマンドを使用してボリュームの状態を表示できます。

スナップショットボリュームを逆再同期するには

- 逆再同期操作を開始するには、`vxsfadm -o rrbegin` を使う
- 逆再同期の変更をコミットするには、`vxsfadm -o rrcommit` を使う
- 逆再同期操作を中止するかまたは取り消して元のデータボリュームに戻すには、`vxsfadm -o rrabort` を使う

メモ: スナップショット操作後に `vxsfadm` を使って Storage Checkpoint、FlashSnap、FileSnap、SOS のオブジェクトを削除し、その後で逆再同期操作を行うと、SFDB は削除済みオブジェクトの一覧を表示しません。この動作はオブジェクトエントリが SFAE リポジトリから削除されるために起きます。

---

p.401 の 図 32-4 を参照してください。

逆再同期パラメータの出力例

```
[oracle@testbox ~]$ vxsfadm -a oracle -s flashsnap --name sfae -o
snapshot
Validating database configuration for third-mirror-break-off snapshot:

    ORACLE_SID                  = RR
    ORACLE_HOME                 =
/home/oracle/app/oracle/product/11.2.0/
dbhome_11203
    APP_MODE                    = online
```

```
    ARCHIVELOG_DEST             = /arch
    SNAPSHOT_ARCHIVE_LOG        = yes
    REVERSE_RESYNC_ARCHIVE_LOG  = no
    REVERSE_RESYNC_RECOVERY     = auto
Database validation successful.
snapshot started at Mon Dec 17 13:27:46 2012.
Putting database in backup mode ...                            Done
Validating database volume layout for third-mirror-break-off snapshot:

    Data volumes ready for snapshot:
        Volume/volume-set datavol of diskgroup datadg mounted on
/data.
    Archivelog volume ready for snapshot:
        Volume/volume-set archvol of diskgroup datadg mounted on
/arch.
    Storage units to be used for snapshot from diskgroup datadg:
        sdl sdk sdm sdj
    SNAPSHOT_VOL_PREFIX         = SNAP_
    SNAPSHOT_DG_PREFIX          = SNAP_
Database volume layout validated successfully.
Creating snapshot volumes for data volumes ...                 Done
Taking database out of backup mode ...                         Done
Flushing online redo logs ...                                  Done
Creating snapshot volume for archivelog volume ...             Done
Copying snapshot information to snapshot volume ...            Done
Creating snapshot diskgroups ...                               Done
Deporting snapshot diskgroups ...                              Done
SNAP_datadg
snaphot ended at Mon Dec 17 13:28:00 2012.

[oracle@testbox ~]$ vxsfadm -a oracle -s flashsnap --name sfae -o
rrbegin
Validating reverse resync volumes ... Done
Importing snapshot diskgroups ... Done
Unmounting primary data volumes ... Done
Mounting snapshot volumes ... Done
Creating storage checkpoint ... Done
Starting database recovery ...
Database started.
SFDB vxsfadm WARNING V-81-0951 Note that this is only a temporary
session
for previewing Reverse Resync changes. Reverse Resync operation will
 be
```

```
complete after performing Reverse Resync Commit operation.
Alternatively,
you can use Reverse Resync Abort operation to discard the changes
and
go back to previous state.

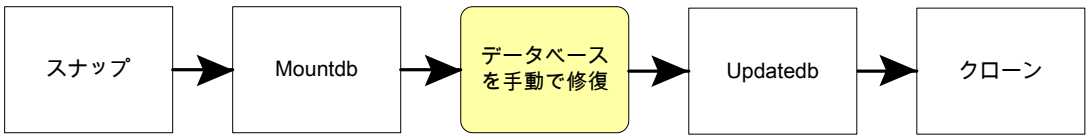
[oracle@testbox ~]$ vxsfadm -a oracle -s flashsnap --name sfae -o
rrabort
Shutting down database RR ...
Database shut down.
Unmounting snapshot volumes ...                                  Done
Restoring storage checkpoint ...                                 Done
Mounting primary data volumes ...                                Done
Deporting snapshot diskgroups ...                                Done

[oracle@testbox ~]$ vxsfadm -a oracle -s flashsnap --name sfae -o
rrcommit
Shutting down database RR ...
Database shut down.
Unmounting snapshot volumes ...                                  Done
Joining snapshot diskgroups to original diskgroups ...           Done
Restoring snapshot volumes onto data volumes ...                 Done
Mounting primary data volumes ...                                Done
Removing storage checkpoint ...                                  Done
Starting database RR ...
Database started.
```

クローンデータベースの手動リカバリ

Oracle データベースでは、データベースの SCN（System Change Number）または特定の時間への回復など、特殊な操作をいくつか実行できます。mountdb 操作を使用して、データベースのリカバリを実行できます。この操作はマウント操作やスナップ操作が完了した後で実行できます。データベースを手動でリカバリするために必要な手順は「図 16-5」で概説されているとおりです。

図 16-5　　データベースの手動によるリカバリ



データベースを手動でリカバリするには

1 ログをリカバリまたは適用せずにデータベースをマウントします。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -o mountdb ¥
-a oracle --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1 --clone_name clone1
Mounting snapshot volumes ... Done
Preparing parameter file for clone database ... Done
Mounting clone database ... Done
```

スナップ操作の後でこのコマンドを実行した場合、ファイルシステムとクローンデータベースをマウントすることになります。アーカイブログは適用されず、データベースでは読み書き操作ができない点に注意してください。基本的な Oracle 回復コマンドを使って、データベースを回復できます。

p.378 の「クローンデータベースの手動による回復」を参照してください。

p.379 の「復元操作の後の手動による回復」を参照してください。

2 リポジトリを更新してデータベースの状態を clone に変更します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -o update_status ¥
-a oracle --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1
```

この手順は、手動での回復の後に状態のクローンを作成するリポジトリで FlashSnap サービスの状態を更新するために実行する必要のあるブックキーピングのためのものです。

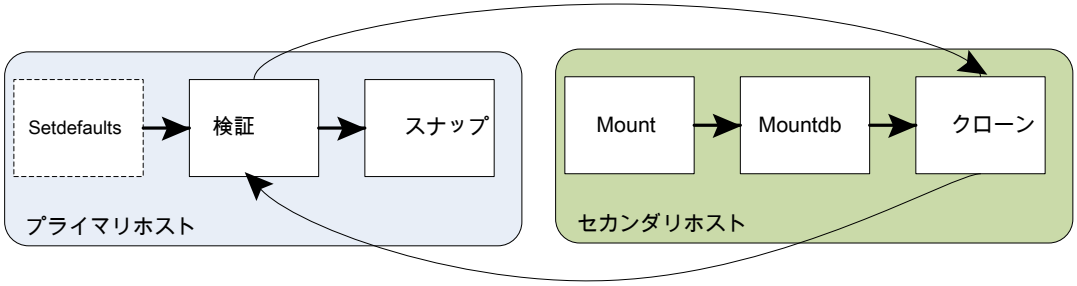
セカンダリホストでのデータベースのクローン作成

vxsfadm コマンドを使用して中間操作を実行することなく最終操作を実行できます。たとえば、通常は「図 16-6」で概説した手順を実行する必要があります。

図 16-6　クローン作成の基本のワークフロー

ただし、vxsfadm を使用して検証から直接クローン作成に進んでセカンダリホストでデータベースのクローン作成をすることができます。「図 16-7」ではこの操作を実行するための順序が概説されています。

図 16-7　　中間操作なしでのクローンの作成



セカンダリホストでデータベースのクローンを作成するには

◆ 次のコマンドを入力します。

```
$ vxsfadm -s flashsnap -a oracle -o clone ¥
--oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--flashsnap_name TEST1 --secondary_host host2
Retrieving snapshot information ...                          Done
Importing snapshot diskgroups ...                            Done
Mounting snapshot volumes ...                                Done
Preparing parameter file for clone database ...              Done
Mounting clone database ...                                  Done
Starting database recovery ...
Clone database clone1 is open.
```

プライマリホスト host1 およびセカンダリホスト host2 の 2 つのホスト設定において、このコマンドは host2 にデータベースのクローンを作成します。

クローンデータベースの SID は clone1 であり、/var/tmp ディレクトリの下にマウントされます。

メモ: ORACLE_HOME と Oracle ユーザーのユーザー ID は、両方のノードで同じになります。その他すべてのパラメータにはデフォルト値が割り当てられます。

DBA として、セカンダリホストのクローンデータベースを SID の異なる別の場所にマウントする必要がある場合、次のコマンドを使います。

```
$ vxsfadm - s flashsnap -a oracle -o clone --oracle_sid sfaedb ¥
--oracle_home /ora_base/db_home -flashsnap_name TEST1 ¥
--secondary_host host2  --clone_path /home/newclonepath ¥
--clone_name newsid
```

17

領域最適化スナップショットの管理

この章では以下の項目について説明しています。

- インスタント領域最適化スナップショットの作成の計画
- インスタントスナップショットを作成するための従来のボリュームの準備
- 共有キャッシュオブジェクトの作成
- 領域最適化スナップショットを使った Oracle データベースのクローンの作成
- FlashSnap スナップショットを使った複数のクローン作成
- クローンデータベースの手動リカバリ

インスタント領域最適化スナップショットの作成の計画

インスタント領域最適化スナップショットを作成するには次の注意事項が適用されます。

- ボリュームの領域最適化インスタントスナップショットを作成する場合、親ボリュームはスナップショットの存在期間中に変更されるため、親ボリュームの元の内容を記録する十分な領域を必要とします。
- 元のボリュームには DCO（バージョン 20 またはそれ以上）と DCO ボリュームがある必用があります。永続 FastResync を有効にする必要があります。
  このリリースでは、これらの必要条件はデフォルトです。これらの設定を変更した場合、または元のボリュームが Storage Foundation の以前のバージョンで作成された場合も、インスタントスナップショットを作成できます。
  p.208 の「インスタントスナップショットを作成するための従来のボリュームの準備」を参照してください。

- ディスクグループ内のボリュームについて複数の領域最適化インスタントスナップショットが必要な場合は、スナップショットごとに別個のキャッシュオブジェクトを作成するよりも、1 つの共有キャッシュオブジェクトをディスクグループ内に作成したほうが便利です。

インスタントスナップショットを作成するための従来のボリュームの準備

インスタントスナップショットでは、元のボリュームが DCO ボリュームを持っていることが必要です。永続的な FastResync を有効にする必要がありますが、これは Storage Foundation 6.0 以降のデフォルトです。元のボリュームが Storage Foundation の旧リリースで作成されている場合は、以下の手順でインスタントスナップショットのボリュームを準備します。この手順は、Persistent FastResync を以前に無効した場合や、`vxsnap unprepare` コマンドをボリュームに対して使用した場合も必要となる場合があります。

インスタントスナップショットの作成のためにレガシーボリュームを準備するには

1 ボリュームがインスタントスナップショットを作成できる DCO ボリュームを持っていることと、FastResync がボリュームで有効になっていることを確認します。

```
# vxprint -g volumedg -F%instant volume
# vxprint -g volumedg -F%fastresync volume
```

両方のコマンドが `on` の値を返す場合は、スナップショットの作成に進みます。

いずれかのコマンドが `off` の値を返す場合は、次の手順に進みます。

2 インスタントスナップショットのボリュームを準備します。

```
# vxsnap [-g diskgroup] prepare volume [regionsize=size] ¥
  [ndcomirs=number] [alloc=storage_attributes]
```

たとえば、ディスクグループ `mydg` にボリューム `myvol` を準備するには、次のコマンドを使います。

```
# vxsnap -g mydg prepare myvol regionsize=128k ndcomirs=2 ¥
  alloc=mydg10,mydg11
```

この例では、DCO オブジェクトと、ディスク `mydg10` および `mydg11` 上に 2 つのプレックスが配置された冗長性のある DCO ボリュームを作成し、さらに `myvol` に関連付けています。領域のサイズもデフォルトの 64 KB から 128 KB に拡張されます。領域のサイズは 16 KB 以上の 2 の累乗の値にする必要があります。値を小さくすると変更マップに必要なディスク領域が増えますが、細分化すればするほど再同期が速くなります。

共有キャッシュオブジェクトの作成

ディスクグループ内のボリュームについて複数の領域最適化インスタントスナップショットを作成する必要がある場合は、スナップショットごとに別個のキャッシュオブジェクトを作成するよりも、1 つの共有キャッシュオブジェクトをディスクグループ内に作成したほうが便利です。

共有キャッシュオブジェクトを作成するには

1. キャッシュオブジェクトの基盤となるキャッシュボリュームに適用する特性を決定します。
   - キャッシュボリュームにはスナップショットの更新の合間に発生した親ボリュームの変更が記録されるため、それに見合ったサイズが必要です。更新間隔が 24 時間の場合の推奨値は、親ボリュームの合計サイズの 10 % です。
   - キャッシュボリュームに冗長性が必要な場合は、ミラー化を行う必要があります。キャッシュボリュームをミラー化すると、必要な領域がミラー数に比例して増大します。
   - キャッシュボリュームをミラー化する場合は、ミラーと同数以上のディスク上に領域が必要になります。親ボリュームと同じディスクは使えません。また、重要なボリュームの I/O 処理効率に影響を与えたり、ディスクグループの分割および結合の操作を妨げることのないよう、適切なディスクを選択する必要があります。
2. 特性を決定したら、`vxassist` コマンドを使って、キャッシュボリュームに使うボリュームを作成します。次の例は、ディスク `disk16` と `disk17` を使ってディスクグループ `mydg` 内に 1 GB のキャッシュボリューム `cachevol` を作成し、そのキャッシュボリュームをミラー化する方法を示しています。

   ```
   # vxassist -g mydg make cachevol 1g layout=mirror ¥
     init=active disk16 disk17
   ```

   属性 `init=active` を指定して、キャッシュボリュームをただちに使えるようにしています。

3 vxmake cache コマンドを使って、前の手順で作成したキャッシュボリュームの上にキャッシュオブジェクトを作成します。

```
# vxmake [-g diskgroup] cache cache_object ¥
  cachevolname=volume [regionsize=size] [autogrow=on] ¥
  [highwatermark=hwmk] [autogrowby=agbvalue] ¥
  [maxautogrow=maxagbvalue]]
```

領域のサイズを指定する場合は、16 KB（16k）以上の 2 の累乗の値にする必要があります。この値を指定しなかった場合、キャッシュの領域のサイズは 64 KB に設定されます。

メモ: キャッシュを共有する領域最適化スナップショットの領域のサイズは、キャッシュに設定した領域のサイズと同サイズかまたはその整数倍にする必要があります。また、元のボリュームの領域のサイズがキャッシュの領域のサイズより小さいと、スナップショットの作成は失敗します。

必要に応じてキャッシュのサイズを拡張できないようにするには、autogrow=off と指定します。特に指定しないかぎり、キャッシュの自動拡張機能は有効になっています。

次の例では、キャッシュボリューム cachevol の上にキャッシュオブジェクト cobjmydg を作成し、キャッシュの領域のサイズを 32 KB に設定して、自動拡張機能を有効にしています。

```
# vxmake -g mydg cache cobjmydg cachevolname=cachevol ¥
  regionsize=32k autogrow=on
```

4 キャッシュオブジェクトの作成が完了したら、次のコマンドを使ってキャッシュオブジェクトを有効にします。

```
# vxcache [-g diskgroup] start cache_object
```

たとえば、キャッシュオブジェクト cobjmydg を起動するには、次のコマンドを実行します。

```
# vxcache -g mydg start cobjmydg
```

領域最適化スナップショットを使った Oracle データベースのクローンの作成

領域最適化スナップショットを使って、次の手順を実行して Oracle データベースのクローンを作成できます。

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.333 の「領域最適化スナップショットの設定パラメータ」を参照してください。

p.335 の「領域最適化スナップショットがサポートされる操作」を参照してください。

詳しくは、vxsfadm-sos（1M）と vxsfadm-oracle-sos（1M）のマニュアルページを参照してください。

領域最適化スナップショットを使って Oracle データベースのクローンを作成するには

1 設定ファイルを作成します。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle ¥
 -o setdefaults --oracle_sid sfaedb --oracle_home
/ora_base/db_home ¥
 --sos_name TEST1 -c dailyconfig
Written config file dailyconfig
```

このコマンドは、**-c** オプションに指定された引数で示された場所にデフォルトの設定ファイルを作成します。パスが指定されていない場合は、現在の作業ディレクトリ内にすべてのパラメータとデフォルト値を含む設定ファイルが作成されます。必要に応じてパラメータを変更できます。

メモ: 指定された名前の設定ファイルがすでにある場合は、一意の設定名を指定するか、異なるパスを指定するか、既存の設定ファイルを削除します。また、破棄操作を実行しても設定ファイルは削除されません。

---

2 設定を検証します。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle -o validate
--oracle_sid ¥
sfaedb --oracle_home /ora_base/db_home --sos_name TEST1
Validating database configuration for space-optimized snapshot:
ORACLE_SID = sfaedb
ORACLE_HOME = /ora_base/db_home
APP_MODE = online
ARCHIVELOG_DEST = /arch/sfaedb
SNAPSHOT_ARCHIVE_LOG = yes
Database validation successful.
Validating database volume layout for space-optimized snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup datadg mounted on /data.
Archivelog volume ready for snapshot:
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
CACHE_OBJECT = datadg:cacheobj
SNAPSHOT_VOL_PREFIX = SNAP_
Database volume layout validated successfully.
```

このコマンドは設定ファイルとデータベース環境を検証します。問題が発生した場合、適切なエラーメッセージが表示されるため、このメッセージを使って問題を修正してから再試行できます。

3 領域最適化スナップショットを作成します。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle ¥
-o snap --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--sos_name TEST1
Validating database configuration for space-optimized snapshot:
ORACLE_SID = sfaedb
ORACLE_HOME = /ora_base/db_home
APP_MODE = online
ARCHIVELOG_DEST = /arch/sfaedb
SNAPSHOT_ARCHIVE_LOG = yes
Database validation successful.
Putting database in backup mode ... Done
Validating database volume layout for space-optimized snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup datadg mounted on /data.
Archivelog volume ready for snapshot:
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
CACHE_OBJECT = datadg:cacheobj
SNAPSHOT_VOL_PREFIX = SNAP_
Database volume layout validated successfully.
Creating snapshot volumes for data volumes ...          Done
Taking database out of backup mode ...                  Done
Flushing online redo logs ...                           Done
Creating snapshot volume for archivelog volume ...      Done
```

このコマンドは、指定したキャッシュオブジェクトを使ってプライマリボリュームから領域最適化スナップショットボリュームを作成します。

メモ: 検証の段階で、必須のパラメータである `--oracle_sid` と `--sos_name` を含め、すべてのパラメータが読み取られてリポジトリに格納されます。

---

4 スナップショットをマウントします。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle ¥
-o mount --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--sos_name TEST1
Mounting snapshot volumes ...                                Done
```

クローンパスを指定する必要がある場合、CLONE_PATH 変数を使います。この変数を指定しない場合、スナップショットはデフォルトで /var/tmp ディレクトリの下にマウントされます。

```
$ vxsfadm -s sos -a oracle -o mount --oracle_sid  sfaedb ¥
--sos_name TEST1 --clone_path /cloneoracle
```

メモ: /cloneoracle ディレクトリが存在しない場合、Oracle ユーザーがこのディレクトリを作成するために必要な権限を持っていることを確認します。

---

5 スナップショットに基づいてデータベースのクローンを作成します。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle -o clone ¥
--oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--sos_name TEST1
Preparing parameter file for clone database ...              Done
Mounting clone database ...                                  Done
Starting database recovery ...
Clone database sfaOjqwF is open.
```

clone_name を指定しなかった場合、自動的に生成されます。

memory_target などのメモリチューニングパラメータを変更する必要がある場合、クローン操作中に変更済みの pfile を指定できます。

レプリケート環境での領域最適化スナップショットも使用できます。

p.316 の「ボリュームスナップショットを VVR と併用する場合の考慮事項」を参照してください。

p.317 の「VVR セットアップでの領域最適化スナップショットの作成」を参照してください。

HA 環境の領域最適化スナップショット

VCS クラスタ内で Oracle データベースをクローンするのに領域最適化スナップショットを使用している場合、切り替える前に次の内容を確認してください。

- Volume エージェントを使用している場合、クローンをマウント解除および破棄してから、キャッシュオブジェクトを無効にします。切り替えの後、データベースのスナップショットを作成し、クローンデータベースを再作成できます。

- クローンを廃棄して、再作成したくない場合は、データベースボリュームの Volume エージェントがある場合に限り、削除します。 代わりに、ディスクグループと Mount エージェントを使用します。

FlashSnap スナップショットを使った複数のクローン作成

オンラインモードのサードミラーブレークオフタイプのスナップショット（データベースインスタンスのオンライン FlashSnap スナップショット）には、データベースインスタンスのクローンを作成するために必要なすべての情報が含まれます。これはクローンデータベースインスタンスを作成するためのテンプレートとして機能します。そのため、1 つの FlashSnap スナップショットを割り当てることで、1 つ以上のクローンインスタンスを作成するためのマスターコピーとして使うことができます。

FlashSnap イメージから作成されるクローンインスタンスは「ゴールデンイメージ」と呼ばれ、マスターイメージ (ゴールデンイメージ) の増分コピーです。これらは操作の FlashSnap イメージに依存します。

驚くような手法で単一のバックアップイメージからの複数のデータベースインスタンスのクローンを作成できます。ここでは、FlashSnap スナップショットの領域最適化スナップショットを作成し、クローンデータベースインスタンスを起動します。

この特定の使用例をサポートするために 2 つの新しい設定パラメータが使われています。

- EXTRA_OBJECTS [short form --extra]
- SOURCE_SNAPSHOT [short form --source]

領域最適化スナップショットを作成するにはキャッシュオブジェクトが必要です。このオブジェクトは FlashSnap スナップショットに含まれているか、スナップショット上に作成されます。EXTRA_OBJECTS を使って FlashSnap スナップショットに含まれているキャッシュオブジェクトを指定します。このパラメータは FlashSnap スナップショットの作成後にキャッシュオブジェクトを作成する場合は必要ありません。

FlashSnap ゴールデンイメージの領域最適化スナップショットを使用してクローンを作成するには

1 ソーススナップショットとして使用するオンラインの FlashSnap スナップショットを作成します。

```
$ vxsfadm -s flashsnap -a oracle --name friday -o snapshot ¥
--oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--extra_objects datadg:cacheobj --extra_objects datadg2:cacheobj2
Validating database configuration for third-mirror-break-off snapshot:
ORACLE_SID                  = sfaedb
ORACLE_HOME                 = /ora_base/db_home
APP_MODE                    = online
ARCHIVELOG_DEST             = /arch/sfaedb
SNAPSHOT_ARCHIVE_LOG        = yes
Database validation successful.
snapshot started at Mon Jan 30 15:05:15 2012.
Putting database in backup mode ...                          Done
Validating database volume layout for third-mirror-break-off snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol2 of diskgroup datadg2 mounted on /data2.
Storage units to be used for snapshot from diskgroup datadg2:
disk_6  Volume/volume-set datavol of diskgroup datadg mounted on /data.
Storage units to be used for snapshot from diskgroup datadg:
disk_3
Archivelog volume ready for snapshot:
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
Storage units to be used for snapshot from diskgroup datadg:
disk_4
SNAPSHOT_VOL_PREFIX         = SNAP_
SNAPSHOT_DG_PREFIX          = SNAP_
Database volume layout validated successfully.
Creating snapshot volumes for data volumes ...               Done
Taking database out of backup mode ...                       Done
Flushing online redo logs ...                                Done
Creating snapshot volume for archivelog volume ...           Done
Copying snapshot information to snapshot volume ...          Done
Creating snapshot diskgroups ...                             Done
Deporting snapshot diskgroups ...                            Done
SNAP_datadg2 SNAP_datadg
snaphot ended at Mon Jan 30 15:05:28 2012.
```

`--extra_objects` オプションを使用すると、スナップショットディスクグループにオリジナルディスクグループからのキャッシュオブジェクトを含めることができます。例では、ディスクグループの datadg と datadg2 からのキャッシュオブジェクトである cacheobj と cacheobj2 とdatadg2 が、スナップショットディスクグループの SNAP_datadg と SNAP_datadg2 に含まれています。このオプションを使わない場

合、スナップショットディスクグループのキャッシュオブジェクトを手動で作成する必要があります。

p.329 の「FlashSnap の設定パラメータ」を参照してください。

2 領域最適化スナップショットを使ってクローンを作成します。

```
$ vxsfadm -s sos -a oracle --name reporting --source-snapshot ¥
friday --prefix RPT -o clone -r primary-host
Validating database configuration for space-optimized snapshot:
ORACLE_SID                  = sfaedb
ORACLE_HOME                 = /ora_base/db_home
APP_MODE                    = online
ARCHIVELOG_DEST             = /arch/sfaedb
SNAPSHOT_ARCHIVE_LOG        = yes
Database validation successful.
Validating source snapshot friday ...                        Done
Validating database volume layout for space-optimized snapshot:
CACHE_OBJECT                = SNAP_datadg2:cacheobj2,SNAP_datadg:cacheobj
SNAPSHOT_VOL_PREFIX         = RPT
Database volume layout validated successfully.
Creating snapshot volumes for data volumes ...               Done
Generated mount path: /var/tmp/IKrlBXRcZ_
Mounting snapshot volumes ...                                Done
Cloning using autogenerated SID ( sfaxwYXY )
Preparing parameter file for clone database ...              Done
Mounting clone database ...                                  Done
Starting database recovery ...
Clone database sfaxwYXY is open.
```

このコマンドは、ゴールデンイメージとして friday という名前の FlashSnap スナップショットを使用してデータベースのクローンを作成します。同じゴールデンイメージを使って複数の領域最適化スナップショットを作成できます。

ゴールデンイメージのスナップショットは、他の FlashSnap スナップショットと同様に、実稼動データベースを実行中のホストとストレージを共有するホストにインポートされます。そのようなオフホストのシナリオでも、FlashSnap ゴールデンイメージからの領域最適化スナップショットを使ってデータベースクローンを作成できます。その場合、`vxsfadm -r` オプションを使う必要があることに注意してください。

実稼働データベースが動作しているホストと異なるホストでクローンを作成できます。このシナリオでは、`vxsfadm` の `-r` オプションを使用してプライマリホストを指定する必要があります。必要なくなったクローンは、`vxsfadm` の `-o delete` オプションを使用して削除することができます。

メモ: ゴールデンイメージとして FlashSnap スナップショットを使用して作成された領域最適化スナップショットが存在する間は、FlashSnap スナップショットは再同期できません。

---

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.333 の「領域最適化スナップショットの設定パラメータ」を参照してください。

p.335 の「領域最適化スナップショットがサポートされる操作」を参照してください。

クローンデータベースの手動リカバリ

Oracle データベースに対しては、データベースをシステム変更番号(SCN)まで、あるいは特定の時間までリカバリするなど、いくつかの特別な操作を実行できます。mountdb 操作を使用して、データベースのリカバリを実行できます。この操作はマウント操作やスナップ操作が完了した後で実行できます。

データベースを手動でリカバリするには

1 ログをリカバリまたは適用せずにデータベースをマウントします。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle ¥
-o mountdb --oracle_sid sfaedb --oracle_home /ora_base/db_home ¥
--sos_name TEST1
Mounting snapshot volumes ...                                Done
```

スナップ操作の後にこのコマンドを実行すると、コマンドはファイルシステムとクローンデータベースをマウントします。アーカイブログは適用されず、データベースは読み取り操作と書き込み操作のために開かれることはありません。基本的な Oracle 回復コマンドを使って、データベースを回復できます。

p.378 の「クローンデータベースの手動による回復」を参照してください。

p.379 の「復元操作の後の手動による回復」を参照してください。

2 リポジトリを更新してデータベースの状態を clone に変更します。

```
$ /opt/VRTSdbed/bin/vxsfadm -s sos -a oracle ¥
-o update_status --oracle_sid sfaedb ¥
--oracle_home /ora_base/db_home --sos_name TEST1
```

この手順は、手動での回復の後に状態のクローンを作成するリポジトリで領域最適化スナップショットサービスの状態を更新するために実行する必要のあるブックキーピングのためのものです。

18

Storage Checkpoint の管理

この章では以下の項目について説明しています。

- Storage Checkpoint について
- リカバリのための Database Storage Checkpoint
- Database Storage Checkpoint の作成
- Database Storage Checkpoint の削除
- Database Storage Checkpoint のマウント
- Database Storage Checkpoint のマウント解除
- Database Storage Checkpoint を使ったデータベースのクローンの作成
- Database Storage Checkpoint からのリストア
- オフラインモード Storage Checkpoint のデータの収集

Storage Checkpoint について

Veritas File System（VxFS）には、Storage Checkpoint 機能が用意されており、特定時刻のファイルシステムの永続的なイメージを瞬時に作成できます。Storage Checkpoint は、コピーオンライト技術を使って最後の Storage Checkpoint またはバックアップの実行後に変更されたファイルシステムブロックのみを認識および保持することにより、I/O オーバーヘッドを大幅に低減します。

p.173 の「コピーオンライト」を参照してください。

Storage Checkpoint は次の機能を提供します。

- 再ブートやクラッシュを通した持続性

- ファイルシステムのメタデータ、ディレクトリ階層、ユーザーデータの保存による、データの即時書き込み機能

Storage Checkpoint は実際はデータオブジェクトで、ファイルシステムにより管理、制御されます。Storage Checkpoint は名前を持つデータオブジェクトであり、作成、削除および名前の変更が可能です。

p.171 の「Storage Checkpoint の動作」を参照してください。

独立した格納領域を必要とするディスクベースのミラー化技術とは異なり、Storage Checkpoint では、ファイルシステムの同じ空き領域内で Storage Checkpoint を使うことにより、使うディスク領域が最小限に抑えられます。

マウントされたファイルシステムの Storage Checkpoint の作成後でも、Storage Checkpoint の論理イメージに影響を与えることなく、ファイルシステムのファイルを引き続き作成、削除および更新できます。Storage Checkpoint では、ファイルシステムの名前空間（ディレクトリ階層）のみならず、ファイルシステムのイメージが取得された時に存在するユーザーデータも保存されます。

Storage Checkpoint には、様々な使い方があります。たとえば、次のような使い方があります。

- テープにバックアップできるファイルシステムの安定したイメージを作成します。
- エンドユーザー独自のファイルが誤って削除されてもそのファイルをリストアできるように、ファイルシステムのマウント済みオンディスクバックアップを作成します。これは、特にホームディレクトリ環境、エンジニアリング環境、電子メール環境で便利な機能です。
- パッチをインストールする前に、問題が発生した時にロールバックできるようにアプリケーションのバイナリーコピーを作成します。
- 従来のテープバックアップに加えて、ファイルシステムのオンディスクバックアップの作成により、高速なバックアップとリストアが行えるようになります。
- Storage Checkpoint を書き込み可能としてマウントすることにより、現在のプライマリファイルセット内の稼動中のデータを危険にさらすことなくプライマリファイルセットのポイントインタイムイメージで新しいソフトウェアをテストします。

リカバリのための Database Storage Checkpoint

Database Storage Checkpoint では、データベースの正確なイメージがすぐに作成され、Database Storage Checkpoint が作成された時点からのデータベースの一貫したイメージが得られます。Database Storage Checkpoint イメージは、コマンドラインインターフェース（CLI）で管理および使用できます。

各 Database Storage Checkpoint はファイルシステムの一貫したポイントインタイムイメージであるため、このオンディスクバックアップに対するリストア機能を実装したアプリケーションが Storage Rollback です。Storage Rollback は、Database Storage

Checkpoint に含まれる、変更されたブロックをプライマリファイルシステムにロールバックすることによって、データベースのリストアを高速化します。

物理的なメディア障害と論理エラーの両方から重要なデータを保護するには、データの冗長性（ミラー化ディスク）と Database Storage Checkpoint を組み合わせて使うことをお勧めします。

Database Storage Checkpoint の利点と制限事項

Database Storage Checkpoint とロールバックには、次の利点があります。

- Database Storage Checkpoint の作成時には、i ノード一覧とプライマリファイルセットのブロックマップ以外、データが格納されていません。ブロックマップは、プライマリファイルシステムの実データを指します。
- Database Storage Checkpoint には i ノード一覧とブロックマップのみが必要で、データはコピーされないため数秒で作成でき、領域もほとんど必要ありません。
- Database Storage Checkpoint は、ブロックレベルで変更された情報を追跡するため、ブロックレベルで段階的にデータベースのバックアップを作成できます。
- Database Storage Checkpoint は、不正に修正されたファイルからデータを回復するのに役立ちます。
- Database Storage Checkpoint をマウントして、標準のファイルシステム操作を実行できます。マウントされた Database Storage Checkpoint は、バックアップ、データ保全調査、ステージングアップグレード、データベースの修正、データ複製ソリューションなど、さまざまなアプリケーションソリューションに使用できます。

Database Storage Checkpoint の制限事項は次のとおりです。

- Database Storage Checkpoint を使えるのは、論理エラー（人為的エラーなど）からリストアする場合に限られます。
- すべてのデータブロックが同一物理デバイス上にあるため、メディア障害が原因で行われるファイルのリストアには Database Storage Checkpoint を使えません。メディア障害が発生した場合は、テープのバックアップや別のメディアに保存されているデータベースファイルのコピーから、データベースをリストアする必要があります。

Database Storage Checkpoint の作成

Storage Foundation の Storage Checkpoint 機能を使って、次の手順を実行することでデータベースの Database Storage Checkpoint を作成できます。

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.338 の「Database Storage Checkpoint の設定パラメータ」を参照してください。

p.340 の「Database Storage Checkpoint のサポートされる操作」を参照してください。

詳しくは、vxsfadm-checkpoint（1M）とvxsfadm-oracle-checkpoint（1M）のマニュアルページを参照してください。

Database Storage Checkpoint を作成するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o create -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ
Putting database in backup mode... Done
Creating Storage Checkpoint XYZ ... Done
Storage Checkpoint XYZ created
Removing the database from backup mode... Done
```

このコマンドは、指定した名前の Oracle データベースの Database Storage Checkpoint を読み取り専用で作成します。*checkpoint_name* パラメータを指定しない場合、名前は自動的に生成されます。

Database Storage Checkpoint の削除

Database Storage Checkpoint を削除するには、次のようにします。

Database Storage Checkpoint を削除するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o delete -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ
Deleting the Checkpoint XYZ...
Deleting Checkpoint from /data... Done
```

このコマンドは、Database Storage Checkpoint XYZ を削除します。

Database Storage Checkpoint のマウント

次のように、Database Storage Checkpoint をマウントできます。

Database Storage Checkpoint をマウントするには

◆ 次のいずれかを実行します。

- 次のコマンドを使って、Database Storage Checkpoint を読み取り専用モードでマウントします。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o mount -a oracle
¥
```

```
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ --mount_path=/tmp/abcd
Mounting Checkpoint to /tmp/abcd... Done
```

このコマンドは、Database Storage Checkpoint を指定したマウントポイントにマウントします。
checkpoint_name パラメータを指定しない場合、名前は自動的に生成されます。

- 読み書き両用モードの Database Storage Checkpoint をマウントするには次のコマンドを使用します。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o mountrw -a oracle
 ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ --clone_path=/tmp/abcd
Creating Storage Checkpoint XYZ_rw_1316516805 ... Done
Storage Checkpoint XYZ_rw_1316516805 created
Mounting Checkpoint to /tmp/abcd... Done
```

このコマンドは、Database Storage Checkpoint XYZ の Storage Checkpoint を作成し、新しく作成した Database Storage Checkpoint を指定したマウントポイントに読み書き両用モードでマウントします。
checkpoint_name パラメータを指定しない場合、名前は自動的に生成されます。

メモ: このコマンドは、元の Database Storage Checkpoint の過去のコピーを保持します。

Database Storage Checkpoint のマウント解除

次のように、Database Storage Checkpoint をマウント解除できます。

Database Storage Checkpoint をマウント解除するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o umount -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ
Unmounting the checkpoint... Done
```

このコマンドは、Database Storage Checkpoint XYZ をマウント解除します。

Database Storage Checkpoint を使ったデータベースのクローンの作成

Database Storage Checkpoint を使用してデータベースクローンを作成するには、次のようにします。

Database Storage Checkpoint のクローンを作成するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o clone -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ --clone_path=/tmp/abcd --clone_name clone1
Creating Storage Checkpoint XYZ_rw_1316523411 ... Done
Storage Checkpoint XYZ_rw_1316523411 created
Mounting Checkpoint to /tmp/abcd... Done
Cloning the Pfile ... Done
Mounting the database... Done
Recovering the clone database... Done
```

このコマンドは、Storage Checkpoint XYZ の Storage Checkpoint を作成し、新しく作成した Storage Checkpoint を指定したマウントポイントに読み書き両用モードでマウントします。

checkpoint_name と *clone_name* を指定しない場合は、自動的に生成されます。

メモ: このコマンドは、もとの Database Storage Checkpoint の過去のコピーを保持します。

Database Storage Checkpoint からのリストア

次のように、Database Storage Checkpoint からデータベースを復元できます。

メモ: 次の操作を実行するには、プライマリデータベースを停止する必要があります。

Database Storage Checkpoint からのデータベースを復元するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o restore -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ
The SCN at the time of taking the checkpoint was : 1170070
Rolling back the application files... Done

You may want to use the following file(s) for application recovery
/var/tmp/hCgJmBm4AT/control01.ctl
/var/tmp/hCgJmBm4AT/binary_controlfile
```

このコマンドは Storage Checkpoint が作成された時、アプリケーションのすべてのデータファイルを PIT に復元します。

メモ: 読み書き両用モードで作成される Database Storage Checkpoint のロールバックはサポートされていません。

復元操作後に、標準の回復手法でデータベースを起動できます。提供される制御ファイルは、Storage Checkpoint の作成時点で存在していた制御ファイルです。このコントロールファイルは、Storage Checkpoint が作成された後にデータベースに何らかの構造の変更があった場合に使用できます。

メモ: 次の操作を実行するには、表領域および表領域内のすべてのデータファイルをオフラインにする必要があります。

表領域内のデータファイルを Database Storage Checkpoint から復元するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o restore -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ --tablespace_rollback newtbs
Rolling back the application files... Done
```

このコマンドは、指定した表領域のすべてのデータファイルを、Database Storage Checkpoint が作成された PIT に復元します。

復元操作後に、標準の回復手法で表領域を回復する必要があります。

メモ: TEMP 表領域はロールバックできません。

メモ: 次の操作を実行するには、復元する必要があるデータファイルをオフラインにする必要があります。

指定したデータファイルを Database Storage Checkpoint から復元するには

◆ 次のコマンドを使います。

```
$ /opt/VRTSdbed/bin/vxsfadm -s checkpoint -o restore -a oracle ¥
--oracle_sid=sfaedb --oracle_home=/ora_base/db_home ¥
--checkpoint_name=XYZ ¥
--datafile_rollback
/data/sfaedb/newf01.dbf,/data/sfaedb/newf02.dbf
Rolling back the application files... Done
```

このコマンドは、指定したすべてのデータファイルを、Database Storage Checkpoint が作成された PIT に復元します。

復元するデータファイルを指定するには、次のいずれかのオプションを使用します。

- datafile_rollback: カンマで区切ったデータファイルの一覧。
- datafile_list_rollback: 改行で区切ったデータファイルのリストが含まれているファイル。

復元操作後に、標準の回復手法でデータファイルを回復する必要があります。

オフラインモード Storage Checkpoint のデータの収集

次のようにオフラインモード Database Storage Checkpoint のデータを収集できます。

メモ: データベースがアップ状態のときのみ、次の操作を実行できます。

データベースがオフライン中に Storage Checkpoint の作成に必要な情報を収集するには

◆ 次のコマンドを使います。

```
$ /opt/VRTS/bin/vxsfadm -s checkpoint -a oracle -o getappdata
Gathering offline data... Done
```

メモ: データ収集操作を実行せずにオフラインの Database Storage Checkpoint を作成しようとすると、Checkpoint 作成が失敗します。

19

FileSnap スナップショットの管理

この章では以下の項目について説明しています。

- FileSnap を使う準備
- FileSnap を使った Oracle データベースのクローンの作成
- FileSnap を使用して作成したデータベースのクローンの破棄
- FileSnap スナップショットの破棄
- FileSnap スナップショットからのデータベースファイルの復元
- FileSnap を使って作成したスナップショットとクローンの一覧表示

FileSnap を使う準備

FileSnap を使用するには、次の内容を満たしていることを確認してください。

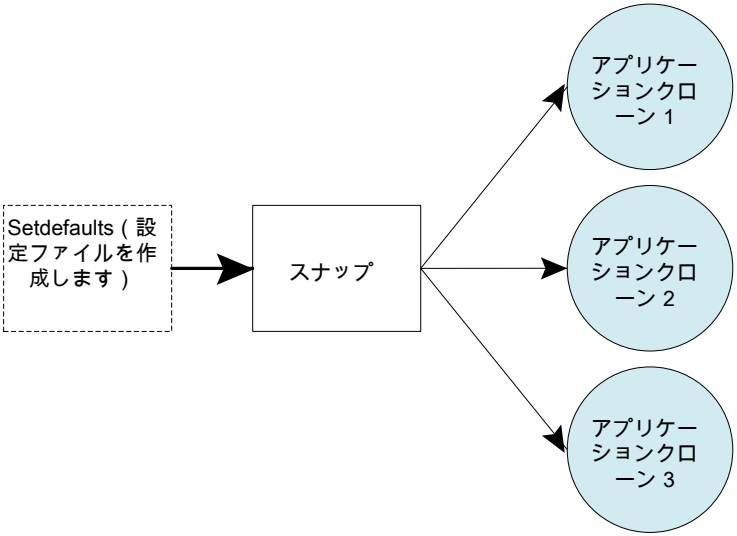
- 下位の VxFS ファイルシステムがディスクレイアウトバージョン 8 以降を使用していること。
- Veritas InfoScale Enterprise とストレージのライセンスが利用可能であること。

一部のファイルシステムがレイアウトバージョン 7 以前を使用している場合、`vxupgrade` ユーティリティを使って、ファイルシステムがオンライン状態のときにバージョン 8 以降にアップグレードします。

FileSnap を使った Oracle データベースのクローンの作成

FileSnap を使って、図 19-1 で概説している手順を実行して Oracle データベースのクローンを作成できます。

図 19-1 クローンの作成 - FileSnap



p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.336 の「FileSnap 設定パラメータ」を参照してください。

p.337 の「FileSnap がサポートされる操作」を参照してください。

詳しくは、`vxsfadm-filesnap`（1M）と `vxsfadm-oracle-filesnap`（1M）のマニュアルページを参照してください。

FileSnap を使って Oracle データベースのクローンを作成するには

1 設定ファイルを作成します。

```
$ vxsfadm -s filesnap -a oracle --oracle_sid sfaedb -oracle_home
 ¥
 '/ora_base/db_home' --filesnap_name 'daily_snap' ¥
 -c dailyconfig -o setdefaults

Written config file dailyconfig
```

このコマンドは、-c オプションで指定された引数が示す場所にデフォルトの設定ファイルを作成します。パスが指定されていない場合は、すべてのパラメータとデフォルト値を含む設定ファイルが現在の作業ディレクトリに作成されます。必要に応じてパラメータを変更できます。

メモ: 同じ名前の設定ファイルがある場合は、異なる設定名を指定するか、異なるパスを指定するか、既存の設定ファイルを削除します。

メモ: 破棄操作を実行しても設定ファイルは削除されません。

---

2 データベースのスナップショットを作成します。

```
$ vxsfadm -s filesnap -a oracle --oracle_sid sfaedb -oracle_home
 ¥
'/ora_base/db_home' --filesnap_name 'daily_snap' -o snap
Validating database configuration for filesnap...
Database validation successful.
ORACLE_SID = sfaedb
ORACLE_HOME = /ora_base/db_home
APP_MODE = online
Validating filesystem configuration for filesnap...
Database filesystem configuration validated successfully.
Starting database online backup mode... Done
Snapshotting files... Done
Stopping database online backup mode... Done
```

このコマンドはすべてのデータファイルの FileSnap スナップショットを作成します。一時ファイル、アーカイブログ、REDO ログはすべて除外されます。

3 データファイルのスナップショットに基づいてプライマリデータベースのクローンを作成します。

```
$ vxsfadm -s filesnap -a oracle --oracle_sid sfaedb -oracle_home
 ¥
'/ora_base/db_home' --filesnap_name 'daily_snap' ¥
 --clone_name clone1 -o clone
Cloning files... Done
Preparing parameter file for clone database... Done
Mounting clone database... Done
Starting clone database recovery... Done
Clone database clone1 is open.
```

FileSnap では、FileSnap スナップショットを使用することにより複数のデータベースのクローンを柔軟に作成できます。FileSnap スナップショットから別のクローンを作成するには、次のコマンドを使用します。

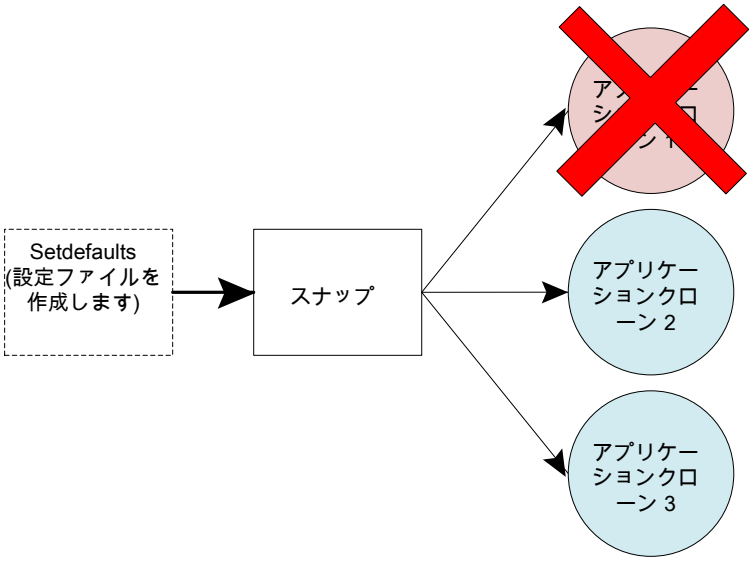
```
$ vxsfadm -s filesnap -a oracle --oracle_sid sfaedb -oracle_home
 ¥
'/ora_base/db_home' --filesnap_name 'daily_snap' ¥
--clone_name clone2 -o clone
Cloning files... Done
Preparing parameter file for clone database... Done
Mounting clone database... Done
Starting clone database recovery... Done
Clone database clone2 is open.
```

メモ: *clone_name* パラメータを指定する必要があります。コマンドラインまたは設定ファイルで指定しない場合でも、自動的に生成されません。

FileSnap を使用して作成したデータベースのクローンの破棄

FileSnap スナップショットを使用して作成したデータベースのクローンは破棄することができます。「図 19-2」では、この操作の概要を示しています。

図 19-2　FileSnap で作成したクローンの破棄



データベースのクローンを破棄するには

◆ 次のコマンドを使用します。

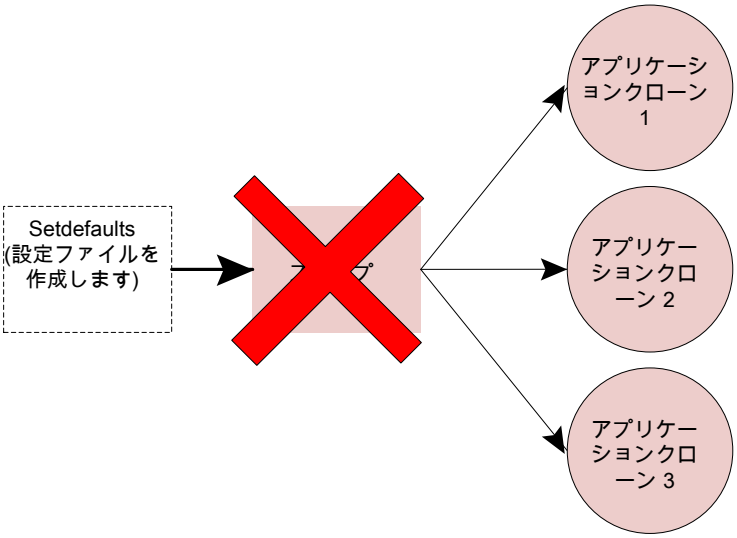
```
$ vxsfadm -s filesnap -a oracle --oracle_sid sfaedb -oracle_home
 ¥
'/ora_base/db_home' --filesnap_name 'daily_snap' ¥
 --clone_name clone1 -o destroyclone
Shutting down clone database... Done
Destroying clone... Done
```

このコマンドは Oracle データベースのクローンを停止し、クローンに関連付けられているファイルを破棄します。この操作によって元のスナップショットが破棄されることはありません。

FileSnap スナップショットの破棄

「図 19-3」で概説される手順を使用してデータファイルの FileSnap スナップショットを破棄できます。

図 19-3　FileSnap スナップショットの破棄



メモ: スナップショットを破棄する前に、スナップショットを使用して作成したすべてのデータベースのクローンを破壊します。

FileSnap スナップショットを破棄するには

◆ 次のコマンドを使用します。

```
$ vxsfadm -s filesnap -a oracle --oracle_sid sfaedb ¥
-oracle_home '/ora_base/db_home' --filesnap_name 'daily_snap' ¥
 -o destroysnap
Destroying snapshot... Done
```

FileSnap スナップショットからのデータベースファイルの復元

現在のデータファイルで、過去の時点で取得した FileSnap スナップショットからデータベースファイルを復元できます。この操作は、データ破損、論理エラーなど、データファイルをスナップショットから復元する必要のあるシナリオで便利です。

メモ: この操作では、プライマリデータベースを停止する必要があります。

メモ: この操作によってデータベースがリカバリすることはありません。

FileSnap スナップショットをリストアするには

◆ 次のコマンドを使用します。

```
$ vxsfadm -s filesnap -a oracle -o restore --oracle_sid ¥
 sfaedb -oracle_home '/ora_base/db_home' --filesnap_name
'daily_snap'
 The SCN at the time of taking the filesnap was : 1174830
 Restoring files... Done

 You may want to use the following file(s) for application
recovery
 /var/tmp/VrBJ8N5XaA/binary_controlfile
```

FileSnap を使って作成したスナップショットとクローンの一覧表示

次の手順を使用して FileSnap により作成されたスナップショットの一覧およびそれらのスナップショットにより作成されたクローンの一覧を表示できます。

データベースインスタンスで作成されたスナップショットおよびクローンの一覧を表示するには

◆ 次のコマンドを使用します。

```
$ vxsfadm -s filesnap -a oracle -o list --oracle_sid sfaedb
The following snapshot(s)/clone(s) are available:
SNAPSHOT NAME     CLONE NAME     STATUS
daily_snap                       Done
                  clone1         Done
                  clone2         Done
```

20

SFHA 環境での Netbackup によるバックアップとリストア

この章では以下の項目について説明しています。

- Veritas NetBackup について
- Veritas NetBackup を使った Oracle のバックアップとリストアについて
- Veritas NetBackup を使った Oracle Disk Manager ファイルのバックアップとリストアについて
- NetBackup を使った Oracle 用 Quick I/O ファイルのバックアップとリストア
- SFHA Solutions 製品環境での NetBackup の使用

Veritas NetBackup について

NetBackup には、クライアントサーバーネットワークのクライアントシステムに格納されているデータベースファイルおよびディレクトリのバックアップ、アーカイブおよびリストア機能が用意されています。NetBackup サーバーソフトウェアは、物理的なバックアップストレージデバイスを管理するプラットフォーム上に存在します。NetBackup サーバーは、自動制御、メディア管理、エラー処理およびスケジュールが行われ、すべてのクライアントバックアップイメージのリポジトリとして機能します。

管理者は、自動的に実行され、介入作業のない完全バックアップおよび増分バックアップのスケジュールを設定できます。前述のバックアップは、NetBackup サーバーによって完全に管理されます。また、管理者が手動でクライアントのバックアップを作成することもできます。クライアントユーザーは各自のクライアントシステムからバックアップ、アーカイブおよびリストアを実行できますが、起動後の操作は NetBackup サーバーに制御されます。

NetBackup は、Storage Foundation Enterprise 製品には付属していませんが、別途購入できます。

Oracle データベースファイルに対する NetBackup の BLI バックアップの動作

BLI バックアップは NetBackup の機能拡張で、Oracle データベースファイルの変更されたデータブロックのバックアップのみを作成します。BLI バックアップでこのバックアップ方式を実行するには、VxFS（Veritas File System）に備わっている Storage Checkpoint 機能を利用します。VxFS は Storage Foundation for Databases ツールを使って利用できます。BLI バックアップにより、データベースのバックアップに必要な時間が短縮され、バックアップ中に転送されるデータ量が軽減されます。これにより、より頻繁にバックアップできるようになるため、より新しいバックアップイメージを保持できるようになります。バックアップからリストアするときの時間は、完全なリストアの完了後に NetBackup が増分バックアップを適用するために必要な時間の分、長くなります。ただし、頻繁に増分バックアップを行うと、適用する REDO ログの数が少なくなるため、データベースリカバリを高速化できます。

BLI バックアップは、データベースが数百 GB または数 TB に及ぶデータベース環境で特に便利です。従来のバックアップ方法によるオフラインデータベースバックアップでは、データベースファイルに変更があると、それがいかに小さい変更でも、データベースファイル全体のバックアップを作成する必要があります。BLI バックアップを使うと、修正されたデータブロックのみのバックアップを作成するだけで済みます。

メモ: BLI バックアップを使うには、データベースが 4、5、6、7 のいずれかのバージョンのレイアウトを使った VxFS ファイルシステム上に存在する必要があります。また、有効な Veritas InfoScale Storage Foundation Enterprise ライセンスも必要です。ファイルシステムのレイアウトバージョンを特定するには、`fstyp -v device` コマンドを使います。詳しくは `vxupgrade`（1M）のマニュアルページを参照してください。

NetBackup for Oracle Advanced BLI Agent をインストールおよび設定し、これを使って Oracle データベースのバックアップを作成する方法について詳しくは、『Veritas NetBackup for Oracle Advanced BLI Agent System 管理者ガイド』を参照してください。

Veritas NetBackup を使った Oracle のバックアップとリストアについて

Veritas NetBackup は、パフォーマンス向上のために、オフライン（コールド）のデータベースバックアップと同様に、週 7 日 24 時間利用可能な必要のあるデータベースのオンライン（ホット）バックアップを提供しています。Veritas NetBackup では、データベースのファイルとディレクトリのバックアップとリストアを行うことができます。完全バックアップや増分バックアップのほか、自動、無人、オンライン、オフラインのデータベースバックアップのスケジュールを設定できます。前述のバックアップは、NetBackup サーバーによって完全に管理されます。また、どの NetBackup クライアントからもデータベースファイル

を手動でバックできます。クライアントのユーザーは、要求に応じてクライアントシステムからデータベースのバックアップとリストアを実行できます。

Veritas NetBackup を使った Oracle Disk Manager ファイルのバックアップとリストアについて

Oracle では、データベースのパフォーマンスを良好に保つために、連続したエクステントレイアウトを持つ Oracle Disk Manager ファイルを割り当てます。Veritas NetBackup を使ってデータベースをバックアップするときには、エクステントの属性が自動的にバックアップされます。データベースファイルをリストアするときには、それらのエクステントの属性を使ってファイルが割り当てられます。バックアップソフトウェアで従来の Oracle RMAN のバックアップ方式を使っている場合は、データファイルも正しいエクステントのレイアウトでリストアされます。

NetBackup を使わない場合、または NetBackup 以外のバックアップソフトウェアで RMAN の「プロキシコピー」バックアップ方式を使っている場合は、エクステントの属性のバックアップが作成されない可能性があります。リストアされたデータファイルのエクステントのレイアウトが正しくなるように、`odmmkfile` コマンドを使って、失われたデータファイルを事前割り当てします。このコマンドは、ファイルをリストアする前にファイル用の連続した領域を事前割り当てします。

`odmmkfile`（1M）マニュアルページを参照してください。

たとえば、100 M のサイズの Oracle データファイルを事前割り当てするには、Oracle データブロックサイズが 8K とすると、`odmmkfile` コマンドを使って次のように入力します。

```
$ /opt/VRTS/bin/odmmkfile -h 8k -s 100m filename
```

NetBackup を使った Oracle 用 Quick I/O ファイルのバックアップとリストア

NetBackup for Oracle を使っている場合は、Oracle RMAN を使って Quick I/O ファイルをバックアップおよびリストアしてください。

NetBackup は、ファイルのバックアップ時にシンボリックリンクをフォローしません。一般的なバックアップ管理アプリケーションは、同じデータを 2 回バックアップしないように、このように設計されています。リンクとそのリンクがポイントするファイルの両方がバックアップ対象のファイルのリストに含まれると、同じファイルが 2 回バックアップされます。

Quick I/O ファイルは、2 つのコンポーネントで構成されています。1 つは、割り当てられた領域を持つ隠しファイルであり、もう 1 つは、その隠しファイルの Quick I/O インターフェースをポイントするリンクです。NetBackup はシンボリックリンクをフォローしないため、バックアップ対象のファイルリストで Quick I/O リンクとそれに対応する隠しファイルの両方を指定する必要があります。

db01 ディレクトリのすべてのファイルとそれらの属性を表示するには、次のコマンドを実行します。

```
$ ls -la /db01

total 2192

drwxr-xr-x    2 root   root   96   Oct 20 17:39 .

drwxr-xr-x    9 root   root   8192   Oct 20 17:39 ..

-rw-r--r--    1 oracle   dba   1048576   Oct 20 17:39

.dbfile

lrwxrwxrwx    1 oracle   dba   22   Oct 20 17:39

dbfile ->¥
.dbfile::cdev:vxfs:
```

上記の例では、シンボリックリンク dbfile と隠しファイル .dbfile をバックアップクラスのファイルリストに含める必要があります。

ディレクトリ内のすべての Quick I/O ファイルのバックアップを作成する場合は、バックアップ対象のディレクトリを指定するだけで、プロセスを単純にすることができます。この場合は、各 Quick I/O ファイルのどちらのコンポーネントも正しくバックアップされます。通常は、ディレクトリ内のすべてのファイルではなく一部のファイルのバックアップを作成するのみの場合を除いて、バックアップ対象のディレクトリを指定するようにしてください。

NetBackup は Storage Foundation と強固に統合されているため、NetBackup は Quick I/O ファイルのエクステント属性をバックアップし、それに従ってリストアを行います。これにより、Quick I/O ファイルのエクステントの予約を保持しつつ、NetBackup を使って Quick I/O ファイルを通常のファイルとしてバックアップおよびリストアできます。この機能がないと、ファイルのリストア時に連続領域の予約が失われるため、パフォーマンスが低下する可能性があります。

Quick I/O ファイルのリストア時に、シンボリックリンクと隠しファイルの両方がすでに存在する場合、NetBackup はバックアップイメージから両方のコンポーネントをリストアします。コンポーネントのいずれかまたは両方が失われている場合、NetBackup は必要に応じて作成または上書きします。

メモ: Oracle のバックアップおよびリストアのユーティリティを使って Quick I/O ファイルをバックアップおよびリストアすることはできません。

一部のバックアップソフトウェアでは、VxFS のエクステント属性をバックアップおよびリストアできない可能性があります。紛失した Quick I/O ファイルをリストアする方法について詳しくは、qio_recreate(1M) オンラインマニュアルページを参照してください。

SFHA Solutions 製品環境での NetBackup の使用

Storage Foundation と NetBackup を統合することで、次の場合に使いやすさと効率性を高めることができます。

- NetBackup マスターサーバーのクラスタ化
- NetBackup を使用した VxVM ボリュームのバックアップとリカバリ
- NetBackup を使用した Oracle データベースのバックアップとリカバリ

NetBackup マスターサーバーのクラスタ化

クラスタ環境で NetBackup マスターサーバーの可用性を高めるには、次の手順を実行します。

NetBackup マスターサーバー、メディア、プロセスの可用性を高めるには

1. NetBackup と Cluster Server のバージョンに互換性があることを確認します。組み合わせについて詳しい情報は、NetBackup クラスタ互換性リストに含まれています。
   - NetBackup 7.x クラスタ互換性: https://www.veritas.com/support/en_US/article.TECH126902 を参照してください
   - NetBackup 6.x クラスタ互換性: https://www.veritas.com/support/en_US/article.TECH43619 を参照してください。
   - NetBackup 5.x クラスタ互換性: https://www.veritas.com/support/en_US/article.TECH29272 を参照してください
   - NetBackup 互換性について詳しくは、https://www.veritas.com/support/en_US/dpp.15145.html を参照してください
2. マスターサーバーをクラスタ化する手順は NetBackup のバージョンによって異なります。指示については、該当する NetBackup マニュアルを参照してください。

   https://sort.veritas.com

VCS リソースと NetBackup プロセスの堅牢性を検証するには

1. Netbackup マスターをオンライン化できるかどうかを検証します。
2. Netbackup マスターをオフライン化できるかどうかを検証します。
3. すべての NetBackup リソースを監視できるかどうかを検証します。

NetBackup を使用した VxVM ボリュームのバックアップとリカバリ

VxVM ボリュームのオブジェクトをバックアップするように NetBackup を有効にするには、次の手順を実行します。この手順では、VxVM ボリュームを使用して IR（インスタントリカバリ）が行えます。

NetBackup を使って VxVM ボリュームのオブジェクトをバックアップするには

1. ディスクが 6 つある VxVM ディスクグループを作成します。ディスクの数はボリュームサイズ、ディスクサイズ、ボリュームレイアウト、およびスナップショット方式によって変わることがあります。

   このテストが実行しているシステムがクラスタ化されたシステムの場合、-s を使用して共有ディスクグループを作成します。

   ```
   # vxdg -s init database_dgdisk1disk2disk3 ¥
   disk4disk5disk6
   ```

2. サイズ 10 GB かディスクの最大サイズのいずれか大きい方の「ミラーストライプ」VxVM ボリュームを作成します。

   ```
   # vxassist -g database_dg make vol_name 10G ¥
   layout=mirror-stripe init=active
   # vxvol -g database_dg set fastresync=on vol_name
   # vxassist -g database_dg snapstart nmirror=1 vol_name
   ```

   メモ: スナップショットには、ミラー、フルサイズインスタント、領域最適化インスタントスナップショットの 3 タイプがあります。例では、IR（インスタントリカバリス）ナップショットを使用します。スナップショット作成について詳しくは、

   『NetBackup Snapshot Client 管理者ガイド 7.6』の 104 から 107 ページを参照してください。

   https://www.veritas.com/support/en_US/article.DOC6459 を参照してください

3. ボリュームでファイルシステムを作成します。
4. ボリュームに VxFS ファイルシステムをマウントします。

   VxVM ボリュームがクラスタ化されたボリュームの場合、「-o クラスタ」オプションが指定された VxFS ファイルシステムをマウントします。

5. VxFS ファイルシステムを希望のレベルまで満たします。たとえば、最大 95% か、使用するファイルシステムに適切なレベルまで満たすことができます。
6. これらのファイルの cksum(1) を格納します。
7. VxFS ファイルシステムをアンマウントします。

8 次の Advanced Client のオプションを有効にします。

- Snapshot Backup を実行します。
- ［拡張スナップショットオプション（Advanced Snapshot Options）］を［vxvm］に設定します。
- ［インスタントリカバリ用にスナップショットを保持する（Retain snapshots for instant recovery）］を有効にします。

9 NetBackup ポリシーで VxVM ボリュームをバックアップします。

『NetBackup Snapshot Client 管理者ガイド 7.6』を参照してください。

https://www.veritas.com/support/en_US/article.DOC6459 を参照してください

NetBackup を使った VxVM ボリュームのリカバリ

VxVM ボリュームのオブジェクトをリカバリするように NetBackup を有効にするには、次の手順を実行します。この手順では、VxVM ボリュームを使用して IR（インスタントリカバリ）を行います。

NetBackup を使って VxVM ボリュームのオブジェクトをリカバリするには

1 VxVM ボリュームをゼロに初期化します。

2 VxVM ボリュームを、初期化された新しい VxVM ボリュームにリカバリします。

3 空の VxVM ボリュームに VxFS ファイルシステムをマウントします。

4 リカバリしたファイルに対して cksum(1) の値を検証します。

NetBackup を使用した Oracle データベースのバックアップ

NetBackup を使用して Oracle データベースをバックアップするには、以下の手順を実行します。

NetBackup の Oracle ポリシーを使用して Oracle 表領域のバックアップを作成するには

1 NetBackup HA または単一の NetBackup サーバーをインストールして設定します。

2 さまざまなレイアウトに基づく Oracle RAC の設定

任意の数のボリュームと任意のボリュームレイアウトを使用して、Oracle データベースを作成します。Oracle データファイルが VxFS ファイルシステムに存在することを確認してください。

3 NetBackup クライアントを Oracle ノードにインストールします。

4 バックアップポリシーを作成します。たとえば、ポリシーは、Flashsnap を使用して Oracle 表領域バックアップを実行するためのポリシーである場合があります。

- ポリシータイプ: Oracle
- スナップショットの方法: Flashsnap

メモ: スナップショットの方法は、Oracle バックアップの場合は省略可能です。

- 代替クライアント: Oracle ホストとデータを共有するホスト

メモ: 代替クライアントは、Oracle バックアップの場合は省略可能です。

代替クライアントオプションを使用する場合、Oracle ホストとストレージを共有するホストが必要です。

- スケジュール: デフォルトアプリケーションバックアップと自動完全バックアップ
- クライアント: Oracle ホスト
- 選択内容: RMAN シェルスクリプトを使用して表領域をバックアップする
- Oracle バックアップポリシーを設定するには、以下のようにします。
  『NetBackup for Oracle 管理者ガイド 7.1: UNIX、Windows、Linux』を参照してください。
  https://www.veritas.com/support/en_US/article.DOC3672 を参照してください。

5 ボリュームでスナップショットの準備をするには、以下のようにします。

『NetBackup Snapshot Client 管理者ガイド 7.1』を参照してください。

https://www.veritas.com/support/en_US/article.DOC3661 を参照してください。

6 データベースの挿入操作の間にデータベースをバックアップするポリシーを使用します。

NetBackup を使用した Oracle データベースの回復

NetBackup を使用して Oracle 表領域回復するには、以下の手順を実行します。

NetBackup の Oracle ポリシーを使用して Oracle 表領域のバックアップを作成し、リカバリするには

1. NetBackup クラスタをフェールオーバーして復元します。
2. RMAN スクリプトを使用して、表領域を元のノードに回復します。

   Oracle 復元用の rman スクリプトを作成するには、以下のようにします。

   『NetBackup for Oracle 管理者ガイド 7.1: UNIX、Windows、Linux』を参照してください。

   https://www.veritas.com/support/en_US/article.DOC3672 を参照してください。
3. 表領域をオンラインにします。
4. 復元したデータベースを確認します。
5. 表内の行数を確認します。

6

Oracle に対するストレージコストの最適化

- 第21章 SmartTier によるストレージの階層化を理解する
- 第22章 SmartTier の設定と管理
- 第23章 Oracle での SmartTier のユースケース
- 第24章 ストレージコストを最適化するためのファイルとデータベースの圧縮
- 第25章 圧縮アドバイザツールの使用

21

SmartTier によるストレージの階層化を理解する

この章では以下の項目について説明しています。

- SmartTier について
- SmartTier for Oracle の動作方法
- 高可用性（HA）環境での SmartTier

SmartTier について

SmartTier は、データ使用上の必要条件に基づいてデータストレージの一致処理を行います。データの一致処理が終了すると、データの使用上の必要条件と、ストレージ管理者またはデータベース管理者（DBA）により定義された他の必要条件に応じてこのデータが再配置されます。

時間の経過とともに保持されるデータが増えるにつれ、一部のデータはしだいに使用頻度が低くなっていきます。使用頻度が低くても、データを保持するには多くのディスク領域が必要です。SmartTier を使うと、データベース管理者が、使用頻度の低いデータをより低速で安価なディスクに移動してデータを管理できるようになります。また、頻繁に利用されるデータをより高速のディスクに保持してすばやく検索できるようにすることも可能です。

ストレージの階層化は、複数のタイプのデータをさまざまなストレージに割り当てることで、パフォーマンスを高速化し、経費を削減する方法です。SmartTier では、ストレージクラスを使って、特定の層を構成するディスクを指定します。ストレージクラスを定義するには、一般的に 2 つの方法があります。

- 処理効率またはストレージが要求されるクラス: 最も使用頻度が高いクラスは、高速で高価なディスクです。定期的には必要とされなくなったデータは、低速で安価なディスクで構成される別のクラスに移動できます。

- 耐障害性を備えたクラス: 各クラスは、非ミラーボリューム、ミラーボリューム、n-way 型ミラーボリュームで構成されます。

  たとえば、通常、データベースはデータ、インデックス、ログで構成されます。データは重要であるため、3 方向のミラーで設定できます。インデックスは重要ですが再作成が可能なため、2 方向のミラーで設定できます。REDO ログおよびアーカイブログを毎日採取する必要はありませんが、これらはデータベースリカバリに必須です。ミラー化することもお勧めします。

SmartTier は、作成したルールに従って異なるストレージ階層からファイルストレージ領域を割り当てることができる VxFS の機能です。SmartTier では、現在より柔軟な方法で階層化されたストレージの管理を行うことができます。ストレージの静的な階層化では、ストレージクラスへのアプリケーションファイルの割り当てを手動で 1 回のみ行いますが、これは長期的に見ると柔軟性に欠けます。通常、階層型ストレージ管理のソリューションでは、アプリケーションのアクセス要求に対する処理を実行する前に、ファイルをファイルシステムの名前空間に戻す必要があり、その結果として遅延とランタイムのオーバーヘッドが発生します。一方、SmartTier では、以下のことが可能です。

- 時間とともにファイルの価値が変わると、最適なストレージ階層にファイルを動的に移動することで、ストレージ資産を最適化します。
- ユーザーまたはアプリケーションのファイルアクセス方法を変更せずに、ストレージ階層間のデータの移動を自動化します。
- 管理者が設定したポリシーに基づいてデータを自動的に移行します。これにより、階層化されたストレージに対する操作要件や、データの移動に関連するダウンタイムが不要になります。

メモ: SmartTier は、以前に Dynamic Storage Tiering（DST）と呼ばれていた機能を拡張し、名前を変更したものです。

SmartTier のポリシーでは、初期のファイルの場所と、既存ファイルが再配置される環境を制御します。これらのポリシーをファイルに適用すると、ファイルシステムのボリュームセットの特定のサブセット（配置クラスと呼ばれる）上でそのファイルを作成し、拡張できます。ファイルは、指定した名前、タイミング、アクセス率、ストレージ容量に関連した条件に一致すると、他の配置クラス内のボリュームに再配置されます。

事前設定ポリシーに加え、必要に応じて、SmartTier で高速または低速のストレージにファイルを手動で移動できます。また、有効なポリシーの一覧表示、ファイル状態の表示、ボリュームの使用状況の表示、ファイルの統計情報の表示などを行うレポートを実行できます。

SmartTier では、Veritas InfoScale Storage Foundation Enterprise 製品 に搭載される 2 つの主要なテクノロジを利用しています。1 つは MVS ファイルシステムです。もう 1 つはファイルシステムで管理されるストレージ内でのポリシーに基づくファイルの自動配置です。MVS ファイルシステムは、2 つ以上の仮想ストレージボリュームを占有するので、単一のファイルシステムを異機種混在の可能性もある複数の物理ストレージデバイス

にまたがるようにすることが可能になります。たとえば、1 つ目のボリュームが EMC Symmetrix DMX のスピンドルに存在し、2 つ目のボリュームが EMC CLARiiON のスピンドルに存在するということも可能です。単一の名前空間を提供することで、ユーザーとアプリケーションは複数のボリュームを透過的に認識します。この MVS ファイルシステムでは各ボリュームが識別されるので、個々のファイルが格納されている場所を制御することが可能です。MVS ファイルシステムをポリシーに基づくファイルの自動配置と組み合わせることで、アプリケーションにもユーザーにもダウンタイムの影響を与えずにデータを自動的に移動する理想的なストレージ階層化機能を実現できます。

データベース環境では、アクセス期間のルールを一部のファイルに適用できます。ただし、たとえばデータファイルによっては、アクセスされるたびに更新されて、アクセス期間のルールを使用できない場合もあります。SmartTier には、ファイル全体だけでなくファイルの一部をセカンダリ階層に再配置する機構も用意されています。

SmartTier を使うには、次の機能でストレージを管理している必要があります。

- VxFS MVS ファイルシステム
- VxVM ボリュームセット
- ボリュームタグ
- ファイルレベルの SmartTier 管理
- サブファイルレベルの SmartTier 管理

VxFS MVS ファイルシステムについて

MVS ファイルシステムは、2 つ以上の仮想ボリュームを占有するファイルシステムです。ボリュームの集合はボリュームセットと呼ばれ、1 つの Veritas Volume Manager（VxVM）ディスクグループに属するディスクまたはディスクアレイ LUN で構成されます。MVS ファイルシステムは、単一の名前空間を示し、これによって複数のボリュームの存在がユーザーとアプリケーションに透過的になります。各ボリュームは管理者用に個別の ID 情報を維持し、これによって各ファイルが配置される場所を制御できます。

この機能は、次の必要条件を満たすファイルシステムでのみ利用できます。

- ディスクグループのバージョンが 140 以上。
- ファイルレベルの SmartTier に対するファイルシステムレイアウトのバージョンが 7 以上。
- サブファイルレベルの SmartTier に対するファイルシステムレイアウトのバージョンが 8 以上。

既存の VxFS システムを VxFS MVS ファイルシステムに変換するには、単一ボリュームをボリュームセットに変換する必要があります。

p.259 の「SmartTier for Oracle の設定」を参照してください。

VxFS ボリュームを管理するには、VxFS ボリューム管理ユーティリティ（fsvoladm ユーティリティ）が使えます。fsvoladm ユーティリティは、指定の Veritas File System へのボリュームの追加、削除、サイズ変更、カプセル化、フラグ設定、フラグ消去、フラグ問い合わせといったさまざまな管理作業を行います。

このユーティティについて詳しくは、`fsvoladm` (1M) のマニュアルページを参照してください。

VxVM ボリュームセットについて

ボリュームセットでは、複数のボリュームを 1 つの論理オブジェクトで表すことができます。ボリュームセットを空にすることはできません。基盤となるボリュームとの間のすべての I/O 処理は、ボリュームセットの I/O インターフェースを経由して実行されます。ボリュームセット機能は、Veritas File System（VxFS）のマルチボリュームの機能を強化します。この機能により、ファイルシステムで、基盤となるボリュームの処理効率や可用性などの様々な特性を最大限に活用できます。たとえば、ファイルシステムメタデータを冗長性の高いボリューム上に保存し、ユーザーデータを処理効率の高いボリュームに保存できます。

ボリュームタグについて

ボリュームタグをボリュームに関連付けることによって、配置クラスの VxVM ボリューム部分を作成します。ファイルを配置する目的で、VxFS は配置クラスのボリュームをすべて同等に扱い、ボリューム間に領域割り当てを分散します。ボリュームには、2 つ以上のタグを関連付けることができます。ボリュームに複数のタグを関連付けた場合、そのボリュームは複数の配置クラスに属し、いずれかの配置クラスに関連する割り当てポリシーと再配置ポリシーの影響を受けます。

警告: 複数のタグ付けは慎重に行ってください。

配置クラスは、MVS ファイルシステムのボリュームセットに存在する特定のボリュームの SmartTier 属性です。この属性は文字列で、ボリュームタグと呼ばれます。

SmartTier ファイルの管理

SmartTier では、VxFS MVS ファイルシステムの管理者は、初期ファイルの場所と既存ファイルが再配置される環境を制御する配置ポリシーを定義することによって、ボリュームセットの個々のボリューム上でファイルの配置を管理できます。これらの配置ポリシーをファイルに適用すると、ファイルシステムのボリュームセットの特定のサブセット（配置クラスと呼ばれる）上でそのファイルを作成し、拡張できます。ファイルは、指定した名前、タイミング、アクセス率、ストレージ容量に関連した条件に一致すると、他の配置クラス内のボリュームに再配置されます。

ファイルベースの移動

- 管理者は、ファイルの新規作成前に、ファイル名拡張子に基づくファイル割り当てポリシーを作成し、データベース作成時には適切な階層の上にデータファイルを作成できます。
- また、管理者は、データベースファイルまたは他のタイプのファイルに対してファイル再配置ポリシーを作成し、ファイルが使われる頻度に基づいてファイルを再配置できます。

SmartTier のサブファイルオブジェクトの管理

SmartTier では、MVS VxFS ファイルシステムの管理者は、ファイルオブジェクトや個々のボリューム上のファイル全体の配置を管理することができます。

サブファイルベースの移動によって、次のことができます。

- 指定したマウントセットの指定したファイルセット内の一連の範囲を、要求に応じて目的の階層セットに移動します。
- 次の目的で、ファイルのセグメントを自動的に移動します。
  - 一連のファイルを監視して、I/O 統計情報を収集します。
  - 必要に応じてクラスタ全体で、統計情報を定期的に収集し、保持します。
  - 目的の階層セットに対する相対的なアクセス頻度に基づいて、登録したファイルセットへの範囲制限を定期的に実行します。
  - それらの範囲の移動履歴を追跡します。

SmartTier for Oracle の動作方法

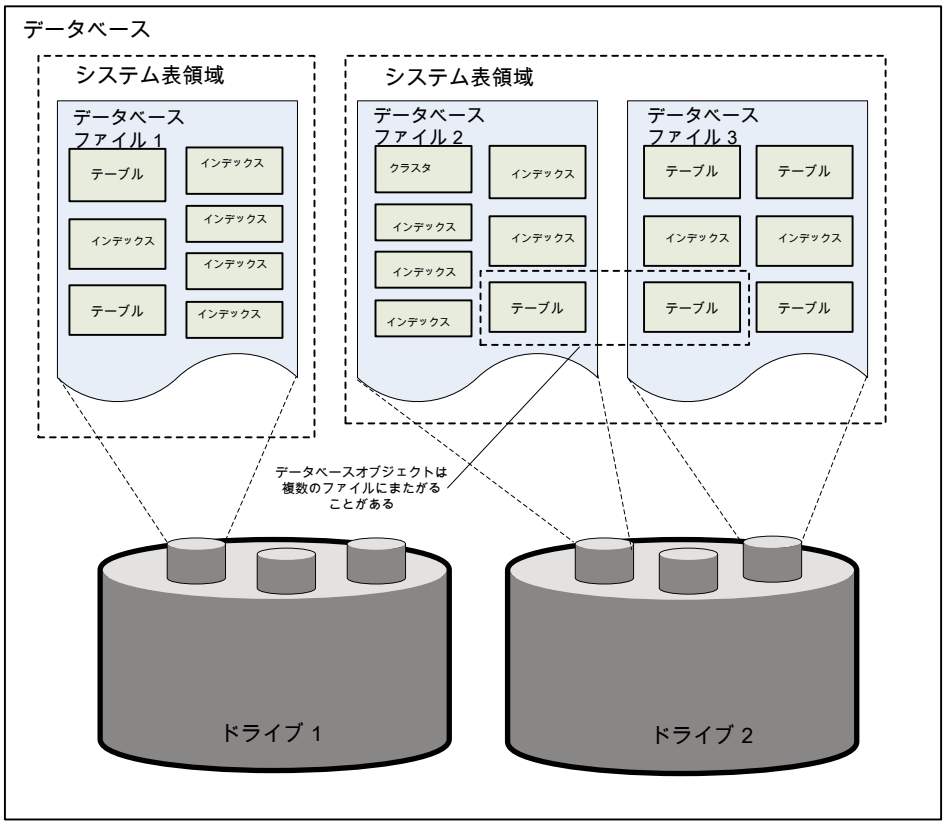
Oracle データベース環境では、アクセス期間のルールを archivelog ファイルと Flashback ファイルに適用することができます。Oracle ではデータベースチェックポイントごとに各データファイルのヘッダーが更新されるので、アクセス期間のルールをデータファイルに使うことはできません。パーティションテーブルの場合は、名前ベースのルールを使って（昨年など）、特定のパーティションに属するファイルをセカンダリのストレージ階層に再配置することができます。ただし、データベースにパーティションテーブルが存在しない場合、現在の方法による再配置は Oracle データベース環境に対応しません。Oracle データベースのファイル再配置を最適化する方法を理解するには、Oracle データベースにオブジェクトが格納されるしくみを知っておく必要があります。

パーティションによる Oracle のデータベースオブジェクトの格納方法の変更

Oracle データベースのデータは、論理的には表領域に格納されていますが、物理的には対応する表領域に関連付けられたデータファイルに格納されています。データベースは、表領域と呼ばれる 1 つ以上の論理ストレージ単位に分割されます。Oracle データ

ベースの表領域は、1 つ以上のデータファイルで構成されます。データファイルは、1 つの表領域および 1 つのデータベースにのみ関連付けることができます。表領域はセグメントと呼ばれる論理ストレージ単位に分割され、さらにセグメントはエクステントに分割されます。エクステントは、データファイル内の連続したブロックの集まりです。テーブルは、Oracle データベースのデータストレージの基本単位です。データは行と列に格納されます。テーブルは、テーブル名（「employees」など）と一連の列で定義されます。通常、パーティションテーブルを除いて、テーブルは単一の表領域内に存在します。パーティションテーブルは、1 つ以上のパーティションを持つテーブルで、各パーティションが別々の表領域に存在することもあります。

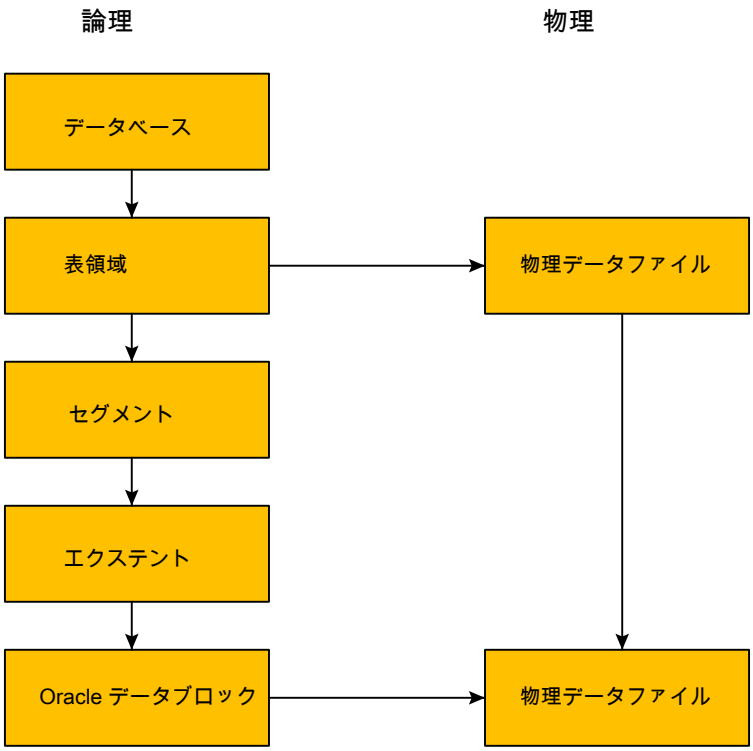
図 21-1 Oracle のデータベースオブジェクトの格納方法



最も細かいレベルの詳細度では、Oracle データベースのデータはデータブロックに格納されます。データブロックは、論理ブロック、Oracle ブロック、ページと呼ばれることもあります。1 つのデータブロックは、ディスク上で決められたバイト数の物理データベース領域に対応します。データベースの論理的な領域の次のレベルは、エクステントです。エクステントは、決められた数の連続したデータブロックであり、表領域内の該当のデータ

ファイルからの特定の種類の情報を格納するために割り当てられます。データベースの論理ストレージで、エクステントの上位レベルはセグメントと呼ばれます。セグメントは一連のエクステントであり、各セグメントは特定のデータ構造に対して割り当てられ、すべてのセグメントが同じ表領域に格納されます。たとえば、各テーブルのデータは独自のデータセグメント内に格納され、各インデックスのデータは独自のインデックスセグメント内に格納されます。テーブルまたはインデックスがパーティション分割されている場合、各パーティションは独自のセグメントに格納されます。Oracle データベースでは、セグメントの領域はエクステント単位で割り当てられます。セグメントの既存のエクステントがいっぱいになると、Oracle データベースはそのセグメントに対して別のエクステントを割り当てます。エクステントは必要に応じて割り当てられるので、セグメントのエクステントはディスク上で連続する場合もありますが、連続しない場合もあります。セグメントおよびそのすべてのエクステントは、単一の表領域に格納されます。表領域内では、1 つ以上のファイルのエクステントをセグメントに含めることができます。つまり、セグメントは複数のデータファイルにまたがることがあります。ただし、各エクステントに含めることができるのは、1 つのデータファイルからのデータだけです。セグメントに新しいエクステントが割り当てられると、その時点で Oracle はエクステントを表領域内のすべてのデータファイルに分散（ストライプ化）しようとします。表領域内の空きデータブロックがなくなった場合は、データファイルを追加することができます。論理オブジェクトと物理ストレージの図解を以下に示します。

図 21-2　　ストレージ上の論理オブジェクト



Oracle には、複数の種類のセグメントがあります。セグメント内のデータの使用パターンは、種類ごとに異なります。ストレージ階層化の再配置に関しては、再配置に適するセグメントの種類を知っておくことが重要です。セグメントの種類を以下に示します。

- データセグメント: テーブルの行を格納するデータブロックが含まれます。データセグメントまたはデータエクステントは、セカンダリのストレージ階層への再配置に最も適した候補です。古いエクステントには、新しいエクステントよりも前に挿入された行が含まれます。セグメントとエクステントのプロパティは、辞書テーブル dba_segments と dba_extents をクエリーすることで利用できます。
- インデックスセグメント: テーブルインデックスのブロックが含まれます。これらのブロックへの I/O が最高速度で読み書きされ、常にプライマリのストレージ階層で行われることが、データベースのパフォーマンスにとって重要です。インデックスセグメントは、表領域またはデータファイル内で 1 つの場所から他の場所へ移動しますが、セカンダリ階層への再配置の候補には適していません。
- ロールバックセグメント: UNDO データが含まれます。UNDO データは、データベースのリカバリにとって重要であり、常にプライマリストレージ上に置く必要があります。

- 一時セグメント: Oracle で必要となるソート操作の中間段階で使用される一時的な作業領域です。一時セグメントの I/O 時間はパフォーマンスに影響を及ぼすため、一時セグメントは常にプライマリストレージ上に置く必要があります。

データベースエクステントのファイル位置

テーブルなどの論理オブジェクトは、データベースエクステントとして格納されます。これらのデータベースエクステントは、該当のファイル内で連続したファイルシステムブロックになります。1 つのファイルを等しい数のオブジェクトに分割しても、データベースエクステントと完全に一致することはありません。これは、Veritas File System の統計情報に関する機能が Oracle データベースに対して機能しない可能性があるためです。1 つのファイルオブジェクトは、複数のデータベースエクステントにまたがる可能性があります。Oracle には、データベースエクステントを（file、offset、length）セットに変換するためのカタログエントリがあります。（file、offset、length）セットを取得すれば、再配置 API を使って、そのエクステントをセカンダリ階層に再配置することができます。

Oracle のカタログテーブル dba_extents と dba_segments を使うと、該当のエクステントに関して（file、offset、length）セットを算出することができます。

指定された dba_segment に関して、dba_segments と dba_extents を組み合わせて、（extent-id、file_id、starting_bloc/offset、length）セットを一覧表示することができます。これによって、データベースエクステントのファイルで正確な位置がわかります。

Oracle の統計情報

Oracle では、さまざまな統計情報を収集し、AWR (Automatic Workload Repository) と呼ばれるデータベースリポジトリ内に格納します。収集される統計情報には、CPU、メモリ、ファイル I/O、ディスクの統計情報があります。AWR は、使用傾向やボトルネックを見つけ出し、パフォーマンスを向上させるのに役立ちます。デフォルトでは、60 分ごとに統計情報のスナップショットが採取されて、AWR に格納されます。スナップショットが AWR で保持される期間はデフォルトで 7 日間です。統計情報の間隔と保持期間は、init.ora を使用して変更できます。Oracle では、統計情報の保持期間を 30 日とすることを推奨しています。要求に応じて追加の統計情報を収集するツールを提供しています。各種のスクリプトとツールによって、AWR をクエリーして、傾向レポートを生成できます。AWR を使うと使用傾向を把握することができるので、使用量の少ないファイルを特定し、それらをセカンダリストレージに再配置できます。

AWR のファイル統計情報を利用することの利点

- AWR のファイル統計情報はデフォルトで利用可能です。追加の設定は必要ありません。
- AWR のファイル統計情報は、データベースのみによって生成される統計情報です。Veritas File System のファイル統計情報には、データベーストランザクションとバックアップやリストアなどの他の I/O の両方が含まれます。

- クラスタ化された Oracle RAC 環境では、AWR のファイル統計情報にはすべてのノードからの I/O が含まれます。各ノードからの情報を収集し、集計する必要はありません。
- AWR のファイル統計情報には履歴も含まれるので、使用傾向を容易に把握できます。

AWR には、セグメントの統計情報など、Oracle オブジェクトレベルの統計情報も含まれます。これらを使うと、使用量の少ないテーブルセグメントを検出することもできます。

Oracle のファイル統計情報の大きい欠点として、ファイルレベル以下の統計情報が得られないという点が挙げられます。

Oracle データベースオブジェクトのストレージの最適化

適切な SmartTier ポリシーを採用し、ストレージのコストを削減するには、頻繁にはアクセスされない Oracle オブジェクトをセカンダリ階層に再配置するための手段が必要です。ファイル全体の再配置が、どのような場合にも有用とは限りません。

再配置の対象として適しているのはデータセグメントだけで、それらは複数のデータファイルにまたがっている可能性があります。新しい行をテーブルに挿入すると、Oracle は要求に応じてテーブルに領域を割り当てます。領域の割り当ては、1 つ以上のエクステントによって行われます。

この割り当てを行うと、特定のデータファイルに次のものが格納されます。

- 複数のテーブルに属する複数のエクステント。
- テーブルに属するエクステントはファイル内で連続していない可能性があります。
- 各データファイルにはヘッダーが含まれ、ヘッダーはデータベースのチェックポイントごとに更新されます。
- ファイルをセカンダリのストレージ階層に再配置することは、1 つ以上のテーブルに属する行またはデータを再配置することを意味します。
- データファイル全体がセカンダリのストレージ階層に再配置される場合、Oracle のチェックポイントのパフォーマンスが影響を受ける可能性があり、いくらかパフォーマンスが低下します。

表領域がローカルで管理されると以下のようになります。

- データファイルの空き領域は、データファイルヘッダーの直後に保持されているビットマップを使って管理されます。
- ビットマップ内の各ビットは、ファイル内のストレージの 1 MB を表しています。
- ビットマップを使う空き領域管理は、辞書によるデータファイルの管理よりも遙かに優れています。
- エクステントがデータファイルの 1 MB を使うように、複数のエクステントが同じファイルから割り当てられます。つまり、各テーブルには 1 MB のストレージが特定のデー

タファイルから割り当てられ、これらすべてのエクステントがデータファイル内で連続しています。

これにより、SmartTier for Oracle を使ったファイルの部分レベルでのストレージの階層化が実用化されます。

表 21-1　推奨される SmartTier for Oracle オプションの使用例

Oracle ファイルの種類	推奨される SmartTier オプション
アーカイブログ	ファイルベースのストレージ階層化
Flashback ログ	ファイルベースのストレージ階層化
パーティション	ファイルベースのストレージ階層化
テーブルの再配置	サブファイルベースのストレージ階層化
ホットデータファイル	サブファイルベースのストレージ階層化
SSD への部分的再配置	サブファイルベースのストレージ階層化

Automatic Workload Repository の統計情報について

Oracle 10g 以降、Oracle は各種の統計情報を収集し、それらを Automatic Workload Repository（AWR）と呼ばれるデータベースリポジトリに格納します。

統計情報には以下のものが含まれます。

- CPU
- メモリ
- ファイル I/O
- ディスクの統計情報

AWR は、使用傾向やボトルネックを見つけ出し、パフォーマンスを向上させるのに役立ちます。

デフォルトでは、AWR スナップショットは以下のように設定されています。

- 60 分ごとに採取されます。
- AWR に格納されます。
- 7 日間保持されます。

統計情報の間隔と保持期間は、init.ora を使用して変更できます。Oracle では、統計情報の保持期間を 30 日とすることを推奨しています。また、要求に応じて追加の統計情報を収集するツールを提供しています。各種のスクリプトとツールによって、AWR をクエ

リーして、傾向レポートを生成できます。AWR を使うとファイルの使用傾向を把握できるので、使用量の少ないファイルをセカンダリストレージ階層に再配置できます。

AWR のファイル統計情報を利用することの利点

- AWR のファイル統計情報はデフォルトで利用可能です。追加の設定は必要ありません。
- クラスタ環境では、AWR のファイル統計情報にはすべてのノードからの IO が含まれます。各ノードからの情報を収集し、集計する必要はありません。
- AWR のファイル統計情報には履歴も含まれるので、使用傾向を容易に把握できます。

AWR には、セグメントの統計情報など、Oracle オブジェクトレベルの統計情報も含まれます。これらを使うと、使用量の少ないテーブルセグメントを検出することもできます。

p.292 の「AWR の統計情報を使った、再配置の可能性があるサブファイルオブジェクトの特定」を参照してください。

高可用性（HA）環境での SmartTier

Cluster Server には、ボリュームセット用の付属エージェントが用意されていません。ボリュームまたはボリュームセットに問題が起きた場合、その問題は DiskGroup リソースと Mount リソースレベルでのみ検出できます。

DiskGroup エージェントは、Veritas Volume Manager（VxVM）ディスクグループをオンライン状態またはオフライン状態にしたり、監視したりします。このエージェントでは VxVM コマンドを使用します。StartVolumes 属性と StopVolumes 属性の値が両方とも 1 の場合、DiskGroup エージェントは、ディスクグループのインポート操作とデポート操作中、ボリュームをオンライン状態にしたり、オフライン状態にしたりします。ボリュームセットを使っている場合、そのボリュームセットを含んでいる DiskGroup リソースの StartVolumes 属性と StopVolumes 属性の値を 1 に設定します。そのボリュームセットにファイルシステムを作成する場合、Mount リソースを使ってボリュームセットをマウントします。

Mount エージェントは、ファイルシステムまたは NFS クライアントのマウントポイントをオンライン状態またはオフライン状態にしたり、監視したりします。

高可用性（HA）環境で SmartTier for Oracle コマンドを使う場合は、クラスタ内の各システムの時刻を同期させる必要があります。同期しない場合、サービスグループのフェールオーバー後、スケジュール設定されたタスクが設定時刻に実行されないことがあります。

詳しくは『Cluster Server 付属エージェントリファレンスガイド』を参照してください。

22

SmartTier の設定と管理

この章では以下の項目について説明しています。

- SmartTier for Oracle の設定
- SmartTier for Oracle を使ったデータベースストレージの最適化
- SmartTier for Oracle を使用したレポートの実行
- SmartTier for Oracle を使ったデータベース環境でのエクステント分散
- SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトのレポートの実行
- SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトの配置の最適化

SmartTier for Oracle の設定

SmartTier for Oracle を使うには、次の必要条件を満たす必要があります。

- Oracle データベースが起動し、すでに稼動している必要があります。
- SmartTier for Oracle コマンドは、Oracle データベース管理者のみが実行できます。

SmartTier for Oracle を設定するには、次のタスクを実行する必要があります。

- SmartTier for Oracle コマンドの必要条件を確認する。
- データベースパラメータを定義する。
- ストレージクラスを設定する。
- SmartTier for Oracle で使うために既存の Veritas File System データベースファイルシステムを Veritas File System MVS ファイルシステムに変換する。
- ボリュームを分類し、タグが下位のディスクの品質を示すようにボリュームにタグを付ける。

- 各クラスの空き領域を表示する。
- 必要に応じてボリュームを追加または削除する。

メモ: SmartTier for Oracle は、以前のリリースで SmartTier for Oracle と呼ばれていた機能を拡張し、名前を変更したものです。

SmartTier for Oracle コマンドの必要条件

データベースのパラメータを定義するには、事前に次のコマンドの必要条件を確認します。

- SmartTier for Oracle コマンドを実行する前に、`dbed_update` コマンドを実行しておく必要があります。データベースファイルを変更する場合もこの `dbed_update` コマンドを実行します。
  SmartTier for Oracle コマンドでは、リポジトリからデータベースの情報を取得するので、リポジトリは最新の情報に維持しておく必要があります。
- SFDB コマンドを使うために環境変数 LD_LIBRARY_PATH を設定する必要はありません。ただし、この環境変数を別の理由で設定する場合は、SFDB コマンドが他のライブラリパスにある同じ名前のライブラリと間違ってリンクしないように、ライブラリパス /opt/VRTSdbed/common/lib を他のライブラリパスの前に指定することをお勧めします。
  SFDB コマンドを使うために環境変数 LIBPATH を設定する必要はありません。ただし、この環境変数を別の理由で設定する場合は、SFDB コマンドが他のライブラリパスにある同じ名前のライブラリと間違ってリンクしないように、ライブラリパス /opt/VRTSdbed/common/lib を他のライブラリパスの前に指定することをお勧めします。
- 高可用性（HA）環境で SmartTier for Oracle コマンドを使う場合は、クラスタ内の各システムの時刻を同期させる必要があります。
- ファイルシステムボリュームと同じディスクグループの MVS ファイルシステムに追加するボリュームを作成します。root ユーザーで次のコマンドを使って各ボリュームの所有者を変更します。

```
# /opt/VRTS/bin/vxedit -g disk_group ¥
set user=oraclevolume
```

- SmartTier for Oracle を実装するマウントポイントの所有者を oracle に変更します。

データベースパラメータの定義

データベース全体のパラメータを定義するには、`dbdst_admin` コマンドを実行します。このコマンドを少なくとも 1 回実行して、SmartTier for Oracle のデータベースパラメータを定義する必要があります。PRIMARY、SECONDARY、BALANCE の 3 つの事前定

義済みストレージクラスが作成されます。パラメータ値は、SFDB リポジトリに格納されます。

maxclass、minclass、statinterval、sweeptime、sweepinterval、purgetime、purgeinterval でパラメータを少なくとも 1 つ設定して、デフォルト値を有効にしてください。少なくとも 1 つのクラスを追加してデフォルトクラスを有効にします。

表 22-1 に、dbdst_admin コマンドのオプションを一覧表示します。

表 22-1 dbdst_admin コマンドオプション

コマンドオプション	説明
-S $ORACLE_SID	ORACLE_SID を指定します。これは、Oracle インスタンスの名前です。
list	データベースのすべての SmartTier for Oracle パラメータ(クラス名、説明など)を一覧表示します。 このオプションは、他のオプションと一緒に使わないでください。
maxclass	データベースに許可されるストレージクラスの最大数を指定します。デフォルト値は 4 です。
minclass	データベースに許可されるストレージクラスの最小数を指定します。デフォルト値は 2 です。
sweepinterval	ファイルを再配置するためにファイルをスイープする間隔を指定します。 デフォルト値は 1 です。この場合、1 日に 1 回実行されます。この値を 0 に設定した場合、すべてのスケジュール設定済みスイープタスクのスケジュールは解除されます。
sweeptime	ファイルのスイープを 1 日に 1 度実行する時刻を指定します。 この値は、24 時間単位で入力し、時間、分の順で指定してください。たとえば、午前 8 時 30 分の場合、08:30、午後 10 時の場合、22:00 と入力します。デフォルト値は 22:00 です。
statinterval	ファイル統計を収集する間隔(分)を指定します。 デフォルト値は 30 です。この場合、30 分間隔で実行されます。この値を 0 に設定した場合、すべてのスケジュール設定済みタスクのスケジュールは解除されます。

コマンドオプション	説明
`purgeinterval`	リポジトリのファイル統計が集計され、パージされるまでの日数を指定します。 デフォルト値は 30 です。パージ間隔を早めに設定することをお勧めします。デフォルト値の場合、最初の 30 日間が過ぎるまで統計を表示できません。
`purgetime`	ファイルのパージを 1 日に 1 度実行する時刻を指定します。 この値は、24 時間単位で入力し、時間、分の順で指定してください。たとえば、午前 8 時 30 分の場合、08:30、午後 8 時の場合、20:00 と入力します。デフォルト値は 20:00 です。
`addclass`	このパラメータを使うと、クラスをデータベースに追加できます。 情報は、class:"description" の形で入力してください。class にクラス名、description に、クラスを説明するための最大 64 文字の文字列(二重引用符で囲む)を入力します。
`rmclass`	このパラメータを使うと、クラスをデータベースから削除できます。データベースに表示されるとおりにクラス名を入力します。

`-o definechunk`
`classname`:
`{128k|256k|512k|1m}`
	任意のストレージクラスのチャンクサイズをバイト単位で定義します。 有効なチャンクサイズは、128 KB、256 KB、512 KB、または 1 MB です。 ストレージクラスのチャンクサイズが指定されている場合、このストレージクラスのファイルはエクステント分散されます。ファイルの各チャンクは、ストレージクラスの別個のボリューム内にあります。 任意のファイルには、ストレージクラスの各コンポーネントボリューム上にほぼ同数のチャンクが存在します。新しいボリュームがストレージクラスに追加されるか、または既存のボリュームがストレージクラスから削除された場合、(dbdst_addvol または dbdst_rmvol を使用して)、ファイルは自動的に再分散されます。

メモ: 特定のデフォルト値を変更しない場合、dbdst_admin コマンドを実行するときに変更しないパラメータを省略できます。入力する必要があるパラメータは、値を変更するパラメータのみです。

データベースパラメータを定義するには

◆ 次のように、dbdst_admin コマンドを使います。

```
dbdst_admin -S $ORACLE_SID -o setup-parameters¥
[,storage_class operations]
        setup-parameters
                maxclass=number,minclass=number,¥
                        statinterval=minutes
                sweeptime=HH:MM,sweepinterval=days
                purgetime=HH:MM,purgeinterval=days
        storage_class operations
                addclass=classname:"description"
                rmclass=classname
                definechunk=classname:128k | 256k | 512k | 1m
dbdst_admin -S $oracle_sid -o list
```

たとえば、データベース PROD に tier1 というクラスを追加し、そのパージ間隔を 1（つまり、1 日分のファイル統計が収集され、その後集計、パージされる）に設定するには、dbdst_admin コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=tier1:"Fast Storage",¥
purgeinterval=1
```

ストレージクラスの設定

データベースパラメータを定義すると、3 つの事前定義済みストレージクラスが作成されます。ユーザーの必要条件に応じてストレージクラスを追加または削除する必要があります。

デフォルトのストレージクラス以外に、ストレージクラスを追加してデータの管理を向上できます。ストレージクラスを追加する前に、次の情報を確認してください。

ストレージクラスを追加するには

◆ 次のように、dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=classname:¥
"description"
```

たとえば、「FAST」というストレージクラスを EMC アレイに作成する場合、次のように dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o addclass=FAST:¥
"fast EMC array"
```

特定のストレージクラスが不要になった場合、そのクラスを削除できます。ストレージクラスを削除する前に、次の情報を確認してください。

メモ: 事前定義済みストレージクラス（PRIMARY、SECONDARY、BALANCE）は削除できません。

ストレージクラスを削除するには

◆ 次のように、dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o rmclass=classname
```

たとえば、「SLOW」というストレージクラスを削除する場合、次のように dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o rmclass=SLOW
```

dbdst_admin コマンドを使って、SmartTier for Oracle のプロパティとストレージクラスのリストを表示できます。

ストレージクラスを表示する前に、次の情報を確認してください。

ストレージクラスを表示するには

◆ 次のように、dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID -o list
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_admin -S VRTS11r2 -o list
TSDB Parameters for VRTS11r2
-----------------------------------
Maxclass        =  4
Minclass        =  2
Stat-Interval   =  30
Sweep Information = 22HH:0MM, Frequency = 1
Purge Information = 20HH:0MM, Frequency = 1


TSDB Storage Classes for VRTS11r2
-----------------------------------
Name = BALANCE : Description = BALANCE Storage Class
Name = PRIMARY : Description = PRIMARY Storage Class
Name = SECONDARY : Description = SECONDARY Storage Class
Name = TIER1 : Description = Fast Storage
```

Veritas File System（VxFS）から VxFS MVS ファイルシステムへの変換

既存の VxFS ファイルシステムを VxFS MVS ファイルシステムに変換するには、単一ボリュームをボリュームセットに変換する必要があります。

`dbdst_convert` コマンドを使ってボリュームセットに変換するとき、もとのボリュームの名前は、新しいボリューム名に変更されます。マウントデバイス名は、新しいボリュームセット名になります。マウントデバイス名を使って新しいボリュームセット名を作成することによって、複数の場所でマウントデバイスの名前を変更する必要がなくなります。

単一ボリュームをボリュームセットに変換するには

1 ボリュームセットに変換する前に、次の条件が満たされていることを確認してください。

前提条件	■ Oracle データベースが停止している必要があります。 ■ 少なくとも 1 つの追加ボリュームを作成してください。 ■ ソースファイルシステムをマウント解除する必要があるためデータベースをシャットダウンする必要があります。

使用に関する注意事項

- SmartTier for Oracle を実装する単一ボリュームのファイルシステムを変換する必要があります。
- dbdst_convert コマンドの実行時にファイルシステムをマウント解除する必要があります。
- ファイルシステムに n 個のボリュームがある場合、1 から n-1 のボリュームはストレージクラス「PRIMARY」に配置され、ボリューム n はストレージクラス「SECONDARY」に配置されます。
- 変換実行時に指定したボリュームは、マウントデバイスと同じディスクグループに属する必要があります。
- ファイルシステムはすべてのノードでマウント解除する必要があります。

2 次のように、dbdst_convert コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_convert -S $ORACLE_SID -M mount_device -v
¥
volume_name,volume_name
```

3 データベースオブジェクトをオンライン状態にします。

たとえば、ボリュームベースの oradata ファイルシステムをマウントデバイス /dev/vx/dsk/oradg/oradata 上で SmartTier for Oracle 対応のボリュームセットファイルシステムに変換する場合は、次のように dbdst_convert コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_convert -S PROD -M ¥
/dev/vx/dsk/oradg/oradata -v new_vol1,new_vol2
```

変換すると、3 つのボリューム（oradata_b4vset、new_vol1、new_vol2）を含む oradata というボリュームセットが生成されます。ファイルシステムには、PRIMARY と SECONDARY という 2 つのストレージクラスが生成されます。ボリュームは次のように割り当てられます。

- PRIMARY ストレージクラスには、ボリューム oradata_b4vset と new_vol1 が配置されます。
- SECONDARY ストレージクラスには、new_vol2 が配置されます。

ボリュームのストレージクラスへの分類

SmartTier ポリシーを作成する前、またはデータを手動で移動する前に、クラスをボリュームに割り当てます。

ボリュームを分類するには

1 クラスをボリュームに割り当てる前に、次の情報を確認してください。

使用に関する注意事項

- Veritas File System ファイルシステムを MVS ファイルシステムに変換しておく必要があります。
- クラスをボリュームに割り当てる前に、dbdst_admin コマンドを使ってストレージクラスを登録する必要があります。
- データベースはオンラインとオフラインのどちらの状態でもかまいません。

2 次のように、dbdst_classify コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID -M mount_device ¥
-v volume_name:class[,volume_name:class]
```

たとえば、「FAST」というクラスをボリューム new_vol1 に割り当てる場合、次のように dbdst_classify コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_classify -S $ORACLE_SID ¥
-M /dev/vx/dsk/oradg/oradata -v new_vol1:FAST
```

ストレージクラスの空き領域の表示

ストレージクラスの空き領域、クラス情報、ボリューム情報を確認するには、dbdst_show_fs コマンドを使います。

表 22-2 に、dbdst_show_fs コマンドのオプションを示します。

表 22-2 dbdst_show_fs コマンドオプション

コマンドオプション	説明
-S $ORACLE_SID	ORACLE_SID を指定します。これは、Oracle インスタンスの名前です。
-o volume	各クラスのボリューム上の空き領域を表示します。

コマンドオプション	説明
-m	マウントポイントを指定します。

ストレージクラスの空き領域を表示するには

1 ストレージクラスの空き領域を表示する前に、次の情報を確認してください。

前提条件

- ファイルシステムがマウントされていることを確認します。
- dbdst_show_fs（1M）のマニュアルページを参照してください。

2 次のように、dbdst_show_fs コマンドを使います。

```
dbdst_show_fs -S $ORACLE_SID -m mount_point
```

次に例を示します。

```
$ dbdst_show_fs -S VRTS11r2 -m /data11r2

MAXCLASS=4
MINCLASS=2
TOTAL CLASS=3

VOLUME NAME                    SIZE      USED    AVAILABLE
-----------                    ----      ----    ---------

CLASS=FAST
datavol1                       1024000   1024000 0

CLASS=MEDIUM
datavol2                       4608000   2347872 2260128

CLASS=PRIMARY
datavol-b4vset                 10240000  3585514 6654486
```

```
dbdst_show_fs -S $ORACLE_SID -m mount_point -o volume
```

次に例を示します。

```
$ dbdst_show_fs -S VRTS11r2 -m /data11r2 -o volume

VOLUME NAME                    SIZE      USED    AVAILABLE  TAG
-----------                    ----      ----    ---------  --
datavol-b4vset                10240000  3585514 6654486    PRIMARY
datavol1                       1024000   1024000 0           FAST
datavol2                      4608000   2347872 2260128    MEDIUM
```

ストレージクラスへの新しいボリュームの追加

dbdst_addvol コマンドを使って、ボリュームをボリュームセットに追加します。

ボリュームセットにボリュームを追加するには

1 ボリュームを追加する前に、次の情報を確認します。

使用に関する注意事項

- ボリュームをストレージクラスに追加するときは、データベースは停止している必要があります。
- データベースファイルシステムをマウントする必要があります。

2 次のように、dbdst_addvol コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID -M mount_device ¥
-v volume_name:class[,volume_name:class]
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_addvol -S VRTS11r2 ¥
-M /dev/vx/dsk/dstdatadgot/datavol -v datavol4:FAST
```

ストレージクラスからのボリュームの削除

必要に応じて、ボリュームセットからボリュームを削除できます。ボリュームを削除するには、dbdst_rmvol コマンドを使います。

1 ボリュームを削除する前に、次の情報を確認します。

使用に関する注意事項

- ストレージクラスからボリュームを削除するときは、データベースは停止している必要があります。
- ファイルシステムのデータを含んでいないボリュームのみを削除できます。

2 次のように、dbdst_rmvol コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_rmvol -S $ORACLE_SID -M mount_device ¥
-v volume_name[,volume_name]
```

次に例を示します。

```
$/opt/VRTS/bin/dbdst_rmvol -S VRTS11r2 ¥
-M /dev/vx/dsk/dstdatadgot/datavol  -v datavol6
```

SmartTier for Oracle を使ったデータベースストレージの最適化

SmartTier for Oracle は複数のツールによってストレージを最適化します。たとえば、手動でファイルまたは表領域を再配置することも、事前設定された SmartTier ポリシーを使うこともできます。

メモ: 次の手順では、Oracle データベース管理者としてコマンドを実行する必要があります。

ストレージパフォーマンス向上のための SmartTier for Oracle のオプション

- ファイルの手動再配置
- 表領域の手動再配置
- テーブルパーティションの手動再配置
- ストレージ階層化を自動化するためのポリシー設定
- レポートの実行
- エクステントの分散
- サブファイルのデータベースオブジェクトのレポートの実行
- サブファイルのデータベースオブジェクトの配置の最適化

SmartTier for Oracle を使ったファイルの手動再配置

表 22-3 に、`dbdst_file_move` コマンドのオプションを示します。

表 22-3 dbdst_file_move コマンドオプション

コマンドオプション	説明
`-o archive[n] | flashback`	移動するアーカイブログまたは Flashback ログを指定します。このオプションと `-f` オプションを同時に使わないでください。
`-o external | datafile`	外部ファイルまたはデータファイルを移動するかどうかを指定します。このオプションは **-f** オプションとともに使います。
`-f listfile`	移動するファイルまたはディレクトリのリストを含んでいるリストファイルを指定します。

コマンドオプション	説明
-c storage_class [:days]	ファイルの移動先ストレージクラスを指定します。days オプションを使った場合、指定した日数の間にアクセスされなかったファイルがそのクラスに移動します。-o datafile オプションを使っている場合、days を指定しないでください。
-R	指定したオブジェクトのポリシーを削除します。

ファイルを手動再配置するには

1 ファイルを再配置する前に、次の情報を確認してください。

使用に関する注意事項　複数のパーティションが同じ表領域に存在することはできません。

2 次のように、dbdst_file_move コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_file_move -S $ORACLE_SID -o datafile ¥
-f listfile -c storage_class:days ¥
[-c storage_class:days]
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_file_move -S VRTS11r2  -o datafile ¥
-f /tmp/external.spec -c MEDIUM

Assign placement policy file successful on filesystem /data11r2
Begin enforcing filesystem /data11r2 ...
Enforce placement policy successful on filesystem /data11r2
```

SmartTier for Oracle を使った表領域の手動再配置

dbdst_tbs_move コマンドを使って、表領域を目的のストレージクラスに移動します。このコマンドは、SFDB リポジトリで表領域ファイル名をクエリーし、ユーザーの当面の必要条件に基づいて 1 回のみ移動を実行します。

表領域を再配置するには

◆ 次のように、dbdst_tbs_move コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_tbs_move -S $ORACLE_SID -t tablespace ¥
-c storage_class
```

ここで、

- *tablespace* には移動する表領域を指定します。
- *storage_class* には、表領域の移動先クラスを指定します。

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_tbs_move -S VRTS11r2  -t MYTBS  -c MEDIUM

Assign placement policy file successful on filesystem /data11r2
Begin enforcing filesystem /data11r2 ...
Enforce placement policy successful on filesystem /data11r2
```

SmartTier for Oracle を使ったテーブルパーティションの手動再配置

dbdst_partition_move を使って、テーブルパーティションを移動します。このコマンドは、テーブルとパーティションの名前を検証するために、データベースをクエリーします。この情報から、データファイルのリストが導出され、目的のクラスへのファイル移動が 1 回のみ実行されます。

テーブルパーティションを再配置する前に、次の情報を確認してください。

前提条件	dbdst_partition_move コマンドを実行するとき、データベースは起動している必要があります。

テーブルパーティションを再配置するには

◆ 次のように、dbdst_partition_move コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_partition_move -S $ORACLE_SID -T table_name
 ¥
-p partition_name -c storage_class
```

このコマンドについての説明は、次のとおりです。

- -T は、テーブル名を示します。
- -p は、パーティション名を示します。
- -c は、テーブルパーティションの移動先のクラスを示します。

たとえば、SALES テーブルの SALES_Q1 パーティションをストレージクラス SLOW に移動する場合、次のように dbdst_partition_move を使います。

```
$ /opt/VRTS/bin/dbdst_partition_move -S $ORACLE_SID -T SALES ¥
-p SALES_Q1 -c SLOW
```

次に別の例を示します。

```
$ /opt/VRTS/bin/dbdst_partition_move -S VRTS11r2  -T EMP1  -p P3  ¥
     -c MEDIUM

Assign placement policy file successful on filesystem /data11r2
Begin enforcing filesystem /data11r2 ...
Enforce placement policy successful on filesystem /data11r2
```

SmartTier for Oracle を使ったストレージ階層化を自動化するためのポリシー設定

dbdst_preset_policy コマンドを使って、ファイルを作成する前にファイル名のパターンに基づいてポリシーを設定します。

表 22-4 に、事前定義ポリシーコマンドのオプションを示します。

表 22-4 dbdst_preset_policy コマンドオプション

コマンドオプション	説明
-d directory	配置ポリシーを適用するディレクトリを指定します。
-e	指定したディレクトリのファイルシステムを実施します。このオプションは、前の実施で起きたエラーが訂正され、再び実施される必要がある場合に使います。
-R	このディレクトリに関連するパターンに基づく配置ポリシーをすべて削除します。
-l	指定したディレクトリに設定されている既存のファイル配置を一覧表示します。
-P pattern_spec	ファイルパターンとクラス割り当てを指定します。このオプションを指定した場合、ファイルは作成されるとすぐに、目的のクラスに自動的に配置されます。既存のファイルと新しく作成されたファイルはすぐに指定したクラスに移動します。
-f pattern file	特定のクラスとパターンを含むファイルを指定します。このパターンを含む新しいファイルは、そのクラスにすぐに配置されます。既存のファイルも同様に配置されます。
-E	dbdst_admin コマンドで指定する sweeptime など、後でスケジュール設定する 1 回のみの移動で、既存のファイルを指定したクラスに移動することを指定します。

事前設定ポリシーを作成するには

◆ 次のように、dbdst_preset_policy コマンドを使います。

```
$ dbdst_preset_policy -S $oracle_sid -d directory ¥
[ -e | -R | -l | -P pattern_spec | -f pattern_file ] [-E]
```

ここで、pattern_spec の形式は「class=pattern,pattern,...[:class=pattern,pattern,...]」です。

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_preset_policy -S VRTS11r2  ¥
     -d /data11r2/VRTS11r2 -P "MEDIUM=*.dbf:FAST=*.log"

Assign placement policy file successful on filesystem /data11r2
Begin enforcing filesystem /data11r2 ...
Enforce placement policy successful on filesystem /data11r2
```

SmartTier for Oracle を使用したレポートの実行

変更された割り当てポリシーを一覧表示するレポートを作成したり、ポリシーを実行した結果として特定の日付範囲に起きた最近の再配置変更を一覧表示する監査レポートを表示したりできます。

修正された割り当てポリシーのリストを作成するには、policy オプションを指定して dbdst_report コマンドを使います。

修正された割り当てポリシーを表示するには

◆ dbdst_report コマンドを使って、修正された割り当てポリシーのリストを表示します。

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o policy

$ /opt/VRTS/bin/dbdst_report -S  VRTS11r1 -o policy
TSDB Database Policy Report
------------------------------------------------------------

Plan Name = preset_dir:/data11r1/VRTS11r1
Plan Type = PRESET POLICY DIRECTORY
Rule Name =  DBED_precreate_ino4_PRIMARY
Directory=/dev/vx/dsk/dstdatadg09/datavol : VRTS11r1
Rule Description =  PRIMARY=MEDIUM=*.log,undo*.dbf

Plan Name = tbs_move:MYTBS
Plan Type = TBS MOVE
Rule Name =  DBED_relocate_once_ino4_MEDIUM
Directory=/dev/vx/dsk/dstdatadg09/datavol : VRTS11r1
Rule Description =  MEDIUM
```

監査レポートには、特定の日付範囲で起きた最近のファイル再配置変更が一覧表示されます。このレポートを表示するには、audit オプションを指定して dbdst_report コマンドを使います。

監査レポートを表示するには

◆ 次のように、dbdst_report コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o audit ¥
startdate=yyyy-mm-dd,enddate=yyyy-mm-dd
```

たとえば、2007 年 1 月 1 日から 2007 年 3 月 1 日までの間の変更を監査レポートに表示するには、次のように dbdst_report コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_report -S $ORACLE_SID -o audit ¥
startdate=2007-01-01,enddate=2007-03-01
```

次に別の例を示します。

```
$ /opt/VRS/bin/dbdst_report -S VRTS11r2 -o audit ¥
startdate=2007-01-01,enddate=2010-08-13

AUDIT DATETIME            AUDIT DESCRIPTION
-------------------------------------------------------
2010-08-13 23:38:48       ADMIN;Added;class name=TIER1
2010-08-13 23:38:48       ADMIN;Modified Successfully;max/min
class=4/2,
 ¥statinterval=30, sweeptime=22:0, purgetime=20:0, swp/prg
freq=1/1
2010-08-13 23:46:48       ADMIN;Added;class name=FAST
2010-08-13 23:46:58       ADMIN;Added;class name=FAST
```

SmartTier for Oracle を使ったデータベース環境でのエクステント分散

データベース環境でパフォーマンスを向上させるには、通常、複数のディスクにストライプ化されたボリュームを使います。ファイルシステムに格納されるデータの量は徐々に増加するため、新しいディスクの形で別の領域を追加する必要があります。

領域を増加する場合、コマンドを実行してボリュームの再レイアウトを実行できます。ただし、たとえば、大容量ボリュームを 4 台にストライプ化されたボリュームから 6 台にストライプ化されたボリュームに変更する場合、古いブロック情報を一時領域に移動し、そのブロックを一時領域から新しいボリュームに書き込む必要があります。これには、長時間の処理が必要です。この問題を解決するために、SFDB ツールには EBFS というエクステント分散ファイルシステムがあります。

エクステント分散ファイルシステムは、個々のボリュームが個々のディスクをまたがってストライプ化されていない MVS ファイルシステムに作成されます。データの可用性を高め

るために、この個々のボリュームをミラー化できます。EBFS 上のファイルシステムには、分散ポリシーと呼ばれる特別な配置ポリシーがあります。 分散ポリシーが適用されると、すべてのファイルが vxrelayout で小さい「チャンク」に分割され、隣接するチャンクが別のボリューム上に配置されるようにチャンクがボリュームに配置されます。デフォルトのチャンクサイズは 1 MB ですが、この値は変更可能です。各ファイルには利用可能なすべてのボリューム上のチャンクが入っているため、EBFS とボリュームセットを構成する個々のボリュームは、サイズとアクセスプロパティが同じである必要があります。

この方法でファイルシステムを設定することによって、ボリュームをストライプ化するときと同じメリットが得られます。

メモ: 既存のファイルシステムを EBFS ファイルシステムに変換することはできません。

データベースパフォーマンスを向上させるためのエクステント分散ファイルシステム

ファイルやファイルシステムに対し、分散の割り当て順序と「チャンク」サイズを指定して、割り当てポリシーを定義できます。これはエクステント分散と呼ばれています。チャンクサイズとは、この割り当てポリシーが適用されているファイルまたはファイルシステムが使用できるエクステントの最大サイズです。チャンクサイズは、分散の割り当て順序のある割り当てポリシーにのみ指定できます。

エクステント分散ポリシーでは、分散の割り当て順序とゼロ以外のチャンクサイズを指定します。 分散の割り当て順序により、ポリシーに指定されているボリューム全体で無作為に割り当てが行われ、各割り当ての最大サイズは指定のチャンクサイズと同じサイズに制限されます。

エクステント分散は、ポリシー内の各ボリュームができる限り均等に使われるようにエクステント割り当てを再分散するなど、ポリシーの実行方法にも拡大適用されます。次のような場合にポリシーが実行されます。

- ポリシーに新しいボリュームが追加され、ファイルに関連付けられているエクステントで、新しいボリュームを含むすべてのボリュームにわたるファイルの再分散が必要な場合。
- ボリュームセットまたはポリシーからボリュームが削除され、削除されたボリュームに常駐するファイルのエクステントをポリシーの他のボリュームに移動する必要がある場合。
- エクステント分散ポリシーがファイルに割り当てられていて、ポリシーに定義されているチャンクサイズの必要条件を満たすために、そのエクステントを再編成する必要がある場合。

エクステント分散ポリシーは、ポリシーに定義されているボリューム内のファイルに属するデータエクステントの分散を図るためのものです。ただし、メタデータについては、エクステント分散ポリシーの割り当てで課される制約はありません。

メモ: 固定エクステントサイズがチャンクサイズ未満の場合、そのエクステントサイズは固定エクステントサイズを乗算した最大サイズが上限とされ、これはチャンクサイズ未満です。固定エクステントサイズがチャンクサイズよりも大きい場合、そのエクステントサイズは固定エクステントサイズになります。

SmartTier for Oracle を使ったエクステント分散ファイルシステムの作成

すべての MVS ファイルシステム(MVFS)は、ストレージ階層でクラスに関連付けられたチャンクサイズがある場合は、エクステント分散ファイルシステムにできます。dbdst_admin コマンドを使うと、クラスにチャンクサイズを定義できます。

たとえば、次の dbdst_admin コマンドは、gold と silver のストレージクラスのチャンクサイズを定義します。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID ¥
-o definechunk=gold:256K
```

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID ¥
-o definechunk=silver:128K
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_admin -S $ORACLE_SID ¥
-o definechunk=TIER1:256K
```

上記のコマンドにより、ストレージクラス gold はエクステント分散されます。

チャンクサイズが一度ストレージ層に定義されると、すべての MVFS をこのストレージ層に分類できます。

たとえば、ボリュームセット /dev/vx/dsk/oradg/ora_vset にファイルシステム /oradata が作成され、ここにデータベースのデータファイルが含まれるとします。また、このデータファイルの名前には拡張子 *.dbf が付いているものとします。この MVFS にストレージクラスを定義するには、次のように dbdst_classify コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_classify  -S $ORACLE_SID ¥
-M /dev/vx/dsk/oradg/ora_vset -v  vol1:GOLD,vol2:GOLD,vol3:GOLD
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_classify -S VRTS11r2 ¥
-M /dev/vx/dsk/dstdatadgot/datavol -v datavol1:FAST,datavol2:MEDIUM
```

MVFS には多数のストレージ層があり、各層のチャンクサイズは異なる場合があることに注意してください。たとえば、上記の例と同じ MVFS で、dbdst_classify コマンドを使ってもう一つのストレージ層を定義できます。

```
$/opt/VRTS/bin/dbdst_classify -S $ORACLE_SID ¥
-M /dev/vx/dsk/oradg/ora_vset -v vol4:silver,vol5:silver
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_classify -S VRTS11r2 ¥
-M /dev/vx/dsk/dstarchdgot/archvol -v archvol1:FAST,archvol2:MEDIUM
```

この時点で、MVFS /oradata にある 2 つのストレージ層は、それぞれのチャンクサイズが異なっています。実際のエクステント分散を作成するには、SmartTier ポリシーを割り当て、実施する必要があります。

ポリシーを定義して実施するには、次のように、dbdst_preset_policy コマンドが使えます。

```
$/opt/VRTS/bin/dbdst_preset_policy -S $ORACLE_SID ¥
-d /oradata -P GOLD=*.dbf:SILVER=*.inx
```

上記の例では、SmartTier ポリシーが作成され、このポリシーは /oradata に割り当てられ、ポリシーが実施されます。*.dbf 形式のすべてのデータファイルは、チャンクサイズ 256K で GOLD 層にエクステント分散され、*.inx 形式のすべての索引ファイルはチャンクサイズ 128K で SILVER 層にエクステント分散されます。

次に別の例を示します。

```
$ /opt/VRTS/bin/dbdst_preset_policy -S VRTS11r2  ¥
-d /data11r2/VRTS11r2 -P "MEDIUM=temp*.dbf:FAST=*.log"

Assign placement policy file successful on filesystem /data11r2
Begin enforcing filesystem /data11r2 ...
Enforce placement policy successful on filesystem /data11r2
```

/oradata MVFS の領域使用率を確認するには、dbdst_show_fs コマンドを使います。次に例を示します。

```
$ /opt/VRTS/bin/dbdst_show_fs -S $ORACLE_SID -m /oradata
```

次に例を示します。

```
$ /opt/VRTS/bin/dbdst_show_fs -S VRTS11r2 -m /data11r2 -o volume

VOLUME NAME                    SIZE      USED    AVAILABLE       TAG
-----------                    ----      ----    ---------       ---
datavol-b4vset                 10240000  3585514 6654486
PRIMARY
datavol1                       1024000   1024000 0               FAST
datavol2                      4608000   2347872 2260128         MEDIUM
```

GOLD 層または SILVER 層でさらに領域が必要な場合、該当するストレージ層に、dbdst_addvol コマンドで新しいボリュームを追加することで領域を追加できます。次に例を示します。

```
$ /opt/VRTS/bin/dbdst_addvol -S $ORACLE_SID ¥
-M /dev/vx/dsk/oradg/ora_vset  -v vol7:GOLD
```

新しいボリュームが追加されると、その直後に SmartTier ポリシーが実施され、新しいボリューム上でのエクステント分散も行われます。この様子を確認するには、dbdst_show_fs コマンドを再実行します。

指定ファイルに関する詳しいエクステント情報を確認するには、fsmap コマンドが使えます。次に例を示します。

```
$ /opt/VRTSvxfs/sbin/fsmap -a /oradata/emp.dbf

$ /opt/VRTS/bin/fsmap -a  /oradata/emp.dbf
```

```
$ fsmap -a /standby/oradata1_1/slave/system01.dbf |more
 Volume   Extent Type  File Offset  Extent Size   File
  vol1      Data           0      262144      /standby/oradata1_1/slave/system01.dbf
  vol2      Data      262144        8192      /standby/oradata1_1/slave/system01.dbf
  vol3      Data      270336      131072      /standby/oradata1_1/slave/system01.dbf
  vol1      Data      401408      262144      /standby/oradata1_1/slave/system01.dbf
  vol2      Data      663552      180224      /standby/oradata1_1/slave/system01.dbf
  vol3      Data      843776       32768      /standby/oradata1_1/slave/system01.dbf
  vol1      Data      876544       49152      /standby/oradata1_1/slave/system01.dbf
  vol2      Data      925696       65536      /standby/oradata1_1/slave/system01.dbf
  vol3      Data      991232      196608      /standby/oradata1_1/slave/system01.dbf
```

SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトのレポートの実行

特定のデータベース、インデックス、パーティションに関する詳細情報を取得するには、dbdst_obj_view コマンドを使います。レポートでは、それぞれのデータベーステーブルまたはインデックスのエクステントがどのストレージクラスに存在するかが示されます。また、MVS ファイルシステムのさまざまなストレージクラス内でデータベースオブジェクトの移動を決定するための情報が得られます。

通常、Oracle の表領域は複数のデータファイルで構成されます。テーブルやインデックスなどのデータベースオブジェクトには、複数のデータファイルから割り当てられたエクステントがあります。通常、1 つのデータファイルには複数のデータベースオブジェクトのエクステントが含まれます。

dbdst_obj_view コマンドは Oracle dba_extents カタログテーブルをクエリーし、各エクステントに対してファイルのオフセットと長さをバイト単位で取得します。このコマンドは、オフセットと長さの情報をファイルシステムの下位レベルの SmartTier コマンドに渡して、エクステントブロックの正確な位置を取得します。

dbdst_obj_view の使用法

```
dbdst_obj_view -S $ORACLE_SID -H $ORACLE_HOME -t table [-s start_extent] [-e end_extent]
dbdst_obj_view -S $ORACLE_SID -H $ORACLE_HOME -i index [-s start_extent] [-e end_extent]
dbdst_obj_view -S $ORACLE_SID -H $ORACLE_HOME -p partition [-s start_extent] [-e
end_extent]
dbdst_obj_view -S $ORACLE_SID -H $ORACLE_HOME -f datafile
```

前提条件
- データベースがオンラインであることが必要です。
- コマンドを Oracle ユーザーとして実行する必要があります。

説明
- SFDB リポジトリが最新の状態であることを確認するために、コマンドを使う前に少なくとも 1 回 dbed_update を実行します。
- サイズの大きいテーブルのレポートを生成する場合は時間がかかります。
- テーブルまたはインデックスの範囲を制限するには、開始エクステントまたは終了エクステントを使います。

Oracle RAC 環境の場合
- データベースを実行している Oracle RAC クラスタのどのノードでも、このコマンドを実行することができます。

表 22-5 に、サブファイルのレポートコマンドのオプションを示します。

表 22-5 dbdst_obj_view コマンドのオプション

コマンドオプション	説明
-S ORACLE_SID	情報の取得元となる Oracle データベースの名前を指定します。
-H ORACLE_HOME	データベースの ORACLE_HOME を指定します。
-t table name	表示レポートを生成する必要のあるテーブルの名前。
-i index name	表示レポートを生成する必要のあるインデックスの名前。
-p partition name	表示レポートを生成する必要のあるパーティションの名前。

コマンドオプション	説明
-s start-extent	テーブルまたはインデックスの開始エクステントを指定します。デフォルトの開始エクステントは 0 です。
-e end-extent	テーブルまたはインデックスの終了エクステントを指定します。このオプションを指定しないと、レポートにはテーブルの最後のエクステントが含められます。

オブジェクトが想定どおりに新しい階層に移動されたことを確認するには

◆ emp テーブル全体が SYMMETRIX に、emp テーブルのエクステント 100 ～ 400 が SSD に、パーティション q1_orders が SSD に再配置されたことを検証するには、次のコマンドを使います。

```
$ /opt/VRTS/bin/dbdst_obj_view -S PROD -H /opt/oracle/home11g ¥
-t emp

File Name             Ext-Id   File-Offset    Length    Tier-Name
---------               ------   -----------     ------
---------
/demo_data/emp1.dbf       1      8462336     1048576    SYMMETRIX
/demo_data/emp1.dbf       2      2170880     1048576     CLARION
/demo_data/emp1.dbf       3      3219456     1048576     CLARION
/demo_data/emp1.dbf       4      9510912     1048576     CLARION
/demo_data/emp1.dbf       5     12656640     1048576     CLARION
/demo_data/emp1.db        6     13705216     1048576    SYMMETRIX
/demo_data/emp1.db        7     14753792     1048576    SYMMETRIX
/demo_data/emp1.dbf       8     15802368     1048576    SYMMETRIX
/demo_data/emp1.db              16850944     1048576    SYMMETRIX
```

SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトの配置の最適化

dbdst_obj_move コマンドを使って、指定したデータベーステーブル、インデックス、またはパーティションのファイルを目的のストレージクラスに移動します。dbdst_obj_move コマンドは Oracle dba_extents カタログテーブルをクエリーし、各エクステントに対してファイルのオフセットと長さをバイト単位で取得します。このコマンドは、これらのオフセットまたは長さの情報を VxFS の下位レベルの SmartTier コマンドに渡して、データベースエクステントを目的のストレージクラスに移動します。

dbdst_obj_move の使用法

```
dbdst_obj_move -S $ORACLE_SID -H $ORACLE_HOME [-v] -t table ¥
     [-s start_extent] [-e end_extent] -c storage_class
dbdst_obj_move -S $ORACLE_SID -H $ORACLE_HOME [-v] -i index ¥
     [-s start_extent] [-e end_extent] -c storage_class
dbdst_obj_move -S $ORACLE_SID -H $ORACLE_HOME [-v] -p partition ¥
     [-s start_extent] [-e end_extent] -c storage_class
```

前提条件
- データベースがオンラインであることが必要です。
- コマンドを Oracle ユーザーとして実行する必要があります。

注意
- SFDB リポジトリが最新の状態であることを確認するために、コマンドを使う前に少なくとも 1 回 `dbed_update` を実行します。

Oracle RAC 環境の場合
- データベースを実行している Oracle RAC クラスタのどのノードでも、このコマンドを実行することができます。

表 22-6 に、サブファイルのオブジェクト移動コマンドのオプションを示します。

表 22-6 dbdst_obj_move コマンドのオプション

コマンドオプション	説明
`-S $ORACLE_SID`	情報の取得元となる Oracle データベースの名前を指定します。
`-H $ORACLE_HOME`	データベースの ORACLE_HOME を指定します。
`-v`	これはオプションの引数です。このオプションが指定されている場合、各エクステントが対象のストレージクラスに移動された後に検証が実行されます。移動先のストレージクラスに十分な空き容量がない場合は、一部のテーブルエクステントが移動されない場合があります。
`-t table name`	表示レポートを生成する必要のあるテーブルの名前。
`-i index name`	再配置が必要なインデックスの名前。
`-p partition name`	再配置が必要なパーティションの名前。同一の表領域内に複数のパーティションを共存させることができます。
`-c storage_class`	テーブルまたはインデックスのエクステントの移動先となるストレージクラスを指定します。

コマンドオプション	説明
-s start_extent	テーブルまたはインデックスの開始エクステントを指定します。デフォルトの開始エクステントは 0 です。
-e end_extent	テーブルまたはインデックスの終了エクステントを指定します。このオプションを指定しない場合、テーブルまたはインデックスのすべてが目的のクラスに移動されます。

emp テーブル全体を SYMMETRIX に再配置するには

1. dbed_update コマンドを実行して、リポジトリが最新であることを確認します。
2. 次のコマンドを実行します。

```
$ /opt/VRTS/bin/dbdst_obj_move -S PROD -H /opt/oracle/home_dir ¥
-t emp -c SYMMETRIX
```

emp テーブルのエクステント 100 ～ 400 を SSD に再配置するには

1. dbed_update コマンドを実行して、リポジトリが最新であることを確認します。
2. 次のコマンドを実行します。

```
$ /opt/VRTS/bin/dbdst_obj_move -S PROD -H /opt/oracle/home_dir ¥
-t emp -s 100 -e 400 -c SSD
```

パーティション q1_orders を SSD に再配置し、再配置を検証するには

1. dbed_update コマンドを実行して、リポジトリが最新であることを確認します。
2. 次のコマンドを実行します。

```
$ /opt/VRTS/bin/dbdst_obj_move -S PROD -H /opt/oracle/home_dir ¥
-v -p q1_orders  -c SSD
```

23

Oracle での SmartTier のユースケース

この章では以下の項目について説明しています。

- Oracle での SmartTier の使用例
- SmartTier for Oracle のファイルベースの使用例
- SmartTier for Oracle サブファイルの使用例

Oracle での SmartTier の使用例

Veritas InfoScale 製品には、ストレージの使用を最適化するためにデータを階層化するストレージ階層化機能である SmartTier が含まれています。

以下の手順例では、次の使用例について説明します。

ファイルレベルの SmartTier for Oracle:

- パーティション分割されたデータと表領域の移行
- アーカイブログと Flashback ログの再配置のスケジュール

サブファイルレベルの SmartTier for Oracle

- AWR の統計情報を使った、再配置の可能性があるサブファイルオブジェクトの特定
- 上位のパフォーマンス階層へのホットデータベースセグメントの再配置
- 低コスト階層への古いテーブルデータの再配置

SmartTier for Oracle のファイルベースの使用例

次のファイルベースの使用例が SmartTier for Oracle でサポートされます。

- パーティション分割されたデータと表領域の移行
- アーカイブログと Flashback ログの再配置のスケジュール

パーティション分割されたデータと表領域の移行

おそらく、データベースに対する複数層ストレージの最も単純な応用は、使用の必要条件の変化に従って、個々の表パーティションを異なる配置クラスに再配置することです。再配置に要する正確な時間が予測できない場合、または再配置が不定期な場合、管理者は、再配置のスケジュールを厳密に定義せずに、必要に応じて表パーティションを再配置できます。

表パーティションの臨時再配置は、たとえば、スポーツ用品やアウトドア用品の小売販売など、季節限定のビジネスに関する売上げと在庫目録を追跡するデータベースで役に立ちます。ある特定の在庫目録（たとえば、夏物用品）の売れる季節が近づくと、来月以降、その季節に適した商品を表すデータベースの表パーティションへのアクセスが頻繁になることが予想されるため、その表パーティションを処理効率の高いストレージに再配置できます。同様に、季節外れの商品を表すパーティションは、そのパーティションに対するアクティビティが少なくなると思われるため、低コストのストレージに再配置できます。

たとえば、スポーツ用品を専門に扱う大規模な小売業者の場合、その売上げのほとんどはカタログによって決まります。製品の詳細は大規模なデータベースに保存され、製品の表はアクティビティの種類に基づいてパーティション分割されます。一部の製品は季節限定で、特定の季節以外はあまり売れません。たとえば、夏にスキー用品が売れることはほとんどありません。季節に基づく移行を実現するには、次の例を参照してください。表 product_tab には、summer と winter という 2 つのパーティションがあることを前提とします。このパーティションはそれぞれ個別のデータファイルにマップされています。

最初に、SmartTier for Oracle を使うようにシステムを設定する必要があります。

fast_storage と slow_storage のストレージクラスを追加するには

◆ 次のように、dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=¥
fast_storage:"Fast Storage for Production DB"

$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=¥
slow_storage:"Slow Storage for Production DB"
```

データベースのファイルシステムを変換し、SmartTier for Oracle で使うボリュームを追加するには

◆ 次のように、dbdst_convert コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_convert -S PROD ¥
-M /dev/vx/dsk/oradg/oradata -v new_vol1,new_vol2,new_vol3
```

ボリュームをストレージクラスに分類するには

◆ 次のように、dbdst_classify コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_classify -S PROD ¥
-M /dev/vx/dsk/oradg/oradata -v new_vol1:fast_storage
```

```
$ /opt/VRTS/bin/dbdst_classify -S PROD ¥
-M /dev/vx/dsk/oradg/oradata -v new_vol2:slow_storage,¥
new_vol3:slow_storage
```

ボリュームを設定した後で、管理者は選択した表領域とパーティションの季節ごとの再配置を指定するファイル配置ポリシールールを定義し、データベースのファイルシステムに割り当てることができます。

冬の初めに、夏のデータを低速ストレージに移動し、冬のデータを高速ストレージに移動するには

◆ 次のように、dbdst_partition_move コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab ¥
-p winter -c fast_storage
```

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab ¥
-p summer -c slow_storage
```

このコマンドによって、product_tab 表の冬のパーティションで構成されるファイルは、配置クラス fast_storage に再配置され、夏のパーティションで構成されるファイルは配置クラス slow_storage に再配置されます。 SmartTier for Oracle は、product_tab の冬のパーティションで構成されるファイルと夏のパーティションで構成されるファイルを識別し、下位の SmartTier サービスを使ってそのファイルをすぐにそれぞれ fast_storage 配置クラスと slow_storage 配置クラスに再配置します。

夏の初めに、冬のデータを低速ストレージに移動し、夏のデータを高速ストレージに移動するには

◆ 次のように、dbdst_partition_move コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab ¥
-p summer -c fast_storage
```

```
$ /opt/VRTS/bin/dbdst_partition_move -S PROD -T product_tab ¥
-p winter -c slow_storage
```

SmartTier for Oracle は、対象のパーティションで構成されるファイルを無条件に目的の配置クラスに再配置する SmartTier ポリシールールを作成します。データベース DST は、このルールをデータベースファイルシステムのアクティブなポリシーにマージし、その結果作成された複合ポリシーをファイルシステムに割り当て、そのポリシーをすぐに実施

して対象ファイルを再配置します。追加されたポリシールールは、アクティブなポリシーのどのルールよりも優先されるため、対象ファイルは、次に dbdst_partition_move コマンドが実行されるまでもとの位置のままです。コマンドが実行された時点で、ルールが削除され、別のルールに置き換わります。

アーカイブログと Flashback ログの再配置のスケジュール

アーカイブログは、データの破損状態からリカバリを実行する主要な機構となっているため、通常、データベースログは高い I/O 処理速度と高いデータ信頼性を持つ高価なストレージに保持されます。アーカイブログの容量が限界に達した場合でさえ、ログは、通常、高速リカバリを目的にオンラインを維持しますが、それも時間が経過すると、このログへの参照はきわめて少なくなる可能性があります。これは、一定期間参照がなければ、アーカイブ化されたデータベースログを低コストのボリュームに再配置できることを意味します。

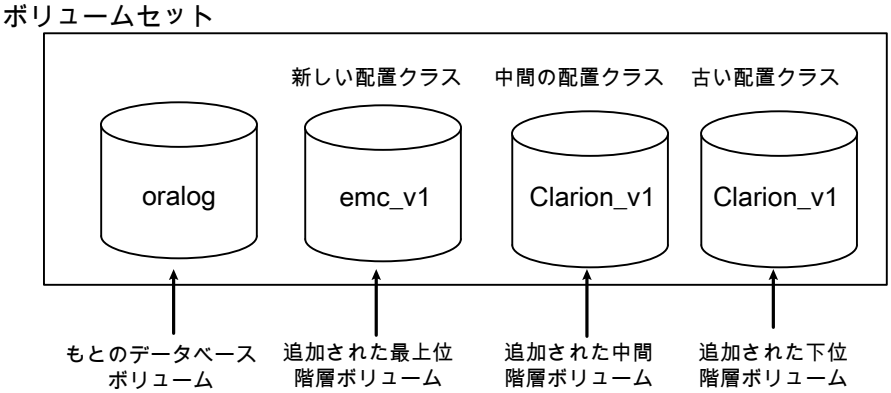
同様に、Storage Foundation の Flashback 技術によって、ログが作成されます。このログは、データベースを以前の状態にリストアすることでデータベースの破損からすばやくリカバリするときに使うことができます。Flashback ログは通常、アーカイブ化されたデータベースログよりも短い期間、維持されます。このログは、使われる場合でも、通常作成されてから数時間の間でしか使われません。一般的な Flashback ログの存続期間は 2、3 日です。

アーカイブログと Flashback ログの使用率の急速な低下は、一定期間参照がないログを定期的に低コストのストレージへ再配置するという配置ポリシーを実施することによって、オンラインストレージの平均コストを削減できることを意味しています。

たとえば、非常に多くのアクティブセッションで大規模な OLTP Oracle データベースを継続して使っているものとします。このとき、このデータベースは、稼動率 99 % 超の状態で 24 時間体制の稼動が必要とされています。データベースは、偶発的なエラーをすばやく訂正するために Flashback 技術を使っています。このデータベースでは 1 日に大量のアーカイブログが生成されます。データベースが何らかの理由で停止した場合、業務上、15 分以内にデータベースをオンラインに戻し、機能を復旧させるという必要条件が発生します。トランザクション時の Oracle ログの切り替え遅延を防止するために、アーカイブログは高速な EMC アレイに作成する必要があります。1 週間を超える古いアーカイブログは、ミッドレンジの Clarion アレイに移動できます。15 日を超える古いアーカイブログは、低速の JBOD ディスクに移動できます。アーカイブログは、30 日後にパージされます。現在の Flashback ログはデータベース管理者によって手動で高速な EMC ストレージに作成されており、2 日後に Clarion ストレージに移動できます。データベース管理者は、1 週間後その Flashback ログを削除します。このようにシステムを設定するには、次の例を参照してください。アーカイブログと Flashback ログは同じファイルシステム /oralog 上に作成されることを前提としています。ファイルシステム上の /oralog/archive1 にアーカイブログが作成され、/oralog/flashback に Flashback ログが作成されます。

図 23-1 は、アーカイブログと Flashback ログの自動再配置と削除に適した 3 層のボリューム設定を表しています。

図 **23-1** アーカイブログと Flashback ログの自動再配置に適したデータベースストレージ設定



この例で実稼動データベースで使われているファイルシステムは、最初は単一ボリューム oralog に存在し、ボリュームとそのボリュームに割り当てられた配置クラスを追加することによって準備する必要があります。

NEW、MEDIUM、OLD のストレージクラスを追加するには

◆ 次のように、dbdst_admin コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=¥
NEW:"EMC Storage for Production DB"

$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=¥
MEDIUM:"Clarion Storage for Production DB"

$ /opt/VRTS/bin/dbdst_admin -S PROD -o addclass=¥
OLD:"JBOD Storage for Production DB"
```

データベースのファイルシステムを変換し、SmartTier for Oracle で使うボリュームを追加するには

◆ 次のように、dbdst_convert コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_convert -S PROD ¥
-M /dev/vx/dsk/oradg/oralog -v emc_v1,clarion_v1,jbod_v1
```

ボリュームをストレージクラスに分類するには

◆ 次のように、dbdst_classify コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_classify -S PROD ¥
-M /dev/vx/dsk/oradg/oralog -v emc_v1:NEW

$ /opt/VRTS/bin/dbdst_classify -S PROD ¥
-M /dev/vx/dsk/oradg/oralog -v clarion_v1:MEDIUM

$ /opt/VRTS/bin/dbdst_classify -S PROD ¥
-M /dev/vx/dsk/oradg/oralog -v jbod_v1:OLD
```

ボリュームの設定が行われると、管理者は、選択したファイルをアクセスの時系列順に再配置してデータベースのファイルシステムに割り当てるようにファイル配置ポリシーのルールを定義できます。

Flashback ログとアーカイブログを定期的に再配置するルールを定義するには

◆ 次のように、dbdst_file_move コマンドを使います。

```
$ /opt/VRTS/bin/dbdst_file_move -S PROD -o flashback -c MEDIUM:2
```

このコマンドによって、2 日間アクセスされなかった Flashback ディレクトリ内のファイルは、MEDIUM ボリュームに再配置されます。

```
$ /opt/VRTS/bin/dbdst_file_move -S PROD -o archive1 -c MEDIUM:7
¥
-c OLD:15
```

このコマンドの実行により、7 日間アクセスがなかった archive1 ディレクトリのファイルは MEDIUM ボリュームに再配置され、15 日間アクセスがなかったファイルは、OLD ボリュームに再配置されます。

SmartTier for Oracle では、これらのコマンドを SmartTier アクセスを時系列順に並べたポリシールールに変換してファイルシステムの配置ポリシーで結合するとともに、変換後のポリシーをそのファイルシステムに割り当てます。デフォルトでは、SmartTier for Oracle で、アクティブポリシーが日単位で施行されます。ポリシーの施行中は、ポリシーの作成に使う dbdst_file_move コマンドで指定したストレージ階層に、新しいルールで対象のファイルが再配置されます。

SmartTier for Oracle サブファイルの使用例

次のサブファイルベースの使用例が SmartTier for Oracle でサポートされます。

- AWR の統計情報を使った、再配置の可能性があるサブファイルオブジェクトの特定
- 上位のパフォーマンス階層へのホットデータベースセグメントの再配置

- 低コスト階層への古いテーブルデータの再配置

AWR の統計情報を使った、再配置の可能性があるサブファイルオブジェクトの特定

Oracle の Automatic Workload Repository（AWR）統計情報を使って、アクティビティレベルなどの主要なプロパティを特定することで、再配置の可能性があるサブファイルオブジェクトを特定できます。Oracle データベースは、自己管理の判断を行うためなど、様々な目的のためにこれらの統計情報を収集します。AWR を有効にして AWR レポートを生成する方法について、以下のガイドラインを参照してください。

AWR 統計情報の収集を有効にするには

◆ 初期設定パラメータ STATISTICS_LEVEL を設定して、データベースとオペレーティングシステムの統計情報の収集レベルを指定します。

- TYPICAL: デフォルトの設定です。この設定により、データベースの自己管理機能に必要なすべての主要な統計情報が収集され、全体的なパフォーマンスが最適化されます。ほとんどの環境に、このデフォルト値が適しています。
- ALL: TYPICAL 設定で収集される統計情報に加えて、さらにその他の統計情報が追加されます。追加の統計情報は、指定時刻の OS 統計情報と計画実行の統計情報です。
- BASIC: Oracle データベースの機能に必要となる重要な統計情報の多くを収集しないようにします。

STATISTICS_LEVEL = typical に設定することをお勧めします。このパラメータは、データベースがオンラインの場合に動的に変更できます。

STATICS_LEVEL を変更するには

◆ 次のコマンドを実行します。

```
SQL> alter system set statistics_level=typical;
```

AWR レポートには、2 つのスナップショット（または 2 つの時点）の間に取得されたデータが表示されます。AWR レポートは、複数のセクションに分割されています。HTML レポートには、セクション間をすばやく移動するためのリンクが用意されています。レポートの内容には、選択した範囲のスナップショットに関するシステムの作業負荷プロファイルが含まれます。

特定範囲のスナップショット ID に対応する HTML レポートまたはテキストレポートを生成するには

1. SQL プロンプトで `awrrpt.sql` スクリプトを実行します。

```
SQL> @$ORACLE_HOME/rdbms/admin/awrrpt.sql
```

2. HTML レポートとテキストレポートのどちらを生成するかを指定します。

```
Enter value for report_type: text
```

3. スナップショット ID の一覧に含める日数を指定します。

```
Enter value for num_days: 2
```

4. 一覧が表示されたら、作業負荷のリポジトリレポートの対象とする開始のスナップショット ID と終了のスナップショット ID を入力するように求められます。

```
Enter value for begin_snap: 150
Enter value for end_snap: 160
```

5. デフォルトのレポート名を使うか、または別のレポート名を入力します。次の例では、デフォルトの名前を使っています。

```
Enter value for report_name:
Using the report name awrrpt_1_150_160
```

作業負荷のリポジトリレポートが生成されます。

AWR レポートを使うと、別のストレージへの移動候補となるテーブルまたはインデックスを特定できます。AWR レポートには、サブファイルレベルの SmartTier に役立つセクションが 2 つあります。目的の階層に対するオブジェクトの移動には、AWR レポートのセグメント統計情報に関する部分を使用できます。TPCC 作業負荷の例を次に示します。セグメント統計情報を使用すると、ホットテーブルまたはホットインデックスを SSD 階層に移動できます。

表 23-1　物理読み取りによるセグメントの例

所有者	表領域名	オブジェクト名	オブジェクトタイプ	物理読み取り	全体に占める比率（%）
TPCC	USERS	ITEM_ID	INDEX	13,551	43.59
TPCC	USERS	ITEM	TABLE	7,718	24.83
TPCC	USERS	CUSTOMER	TABLE	4,403	14.16
TPCC	USERS	STOCK	TABLE	3,692	11.88

所有者	表領域名	オブジェクト名	オブジェクトタイプ	物理読み取り	全体に占める比率(%)
TPCC	USERS	ORDERS	TABLE	352	1.13

上記のセグメント例の ITEM_ID と ITEM は、上位のパフォーマンス階層への再配置で最も可能性のある 2 つの候補です。これらを再配置するコマンドは、次のようになります。

```
$ dbdst_obj_move -S $ORACLE_SID -H $ORACLE_HOME ¥
-i ITEM_ID -c SSD_TIER
```

```
$ dbdst_obj_move -S $ORACLE_SID -H $ORACLE_HOME ¥
-t ITEM -c SSD_TIER
```

`dbdst_obj_move` コマンドの使用方法については、

p.283 の「SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトの配置の最適化」を参照してください。

上位のパフォーマンス階層へのホットデータベースセグメントの再配置

データファイル全体の再配置を実施できない場合は、データベースのホットセグメントまたはホットオブジェクトを上位のパフォーマンスストレージに再配置できます。

以下に示す手順の値に関する例については、表 23-1 を参照してください。

例に示すコマンドは、dba_segments と dba_extents をクエリーして、指定されたテーブルまたはインデックスのエクステントのリストを準備します。エクステントの情報から、(file、offset)を含むリストが生成されます。このリストは、サブファイルの実施を行うために、`fsppadm` に渡されます。

上位のパフォーマンス階層にホットデータベースセグメントを再配置するには

1. Oracle AWR レポートを使って、データベースで最もアクティブなホットオブジェクトまたはホットセグメントを特定します。
2. それらを上位のパフォーマンス階層に再配置するために必要なサイズを決定します。
3. SFDB リポジトリが最新の状態であることを確認するために、`dbed_update` を実行します。
4. テーブル emp を *SSD* 階層に移動するコマンドを実行します。

```
$ dbdst_obj_move  -S PROD -H /home_dir -t EMP -c SSD
```

以下に示す例のコマンドは、指定されたテーブルまたはインデックスが存在する一連のデータファイルを表示します。基になるファイルシステムが MVS ファイルシステムである場合は、対応するボリュームと階層名が表示されます。

データベースセグメントが正しい階層に再配置されていることを確認するには

◆ 再配置されたテーブルを表示するコマンドを実行します。

```
$ dbdst_obj_view  -S PROD -H /home_dir -t EMP
```

dbdst_obj_move コマンドの使用方法については、

p.283 の「SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトの配置の最適化」を参照してください。

低コスト階層またはセカンダリ階層への古いテーブルデータの再配置

TELCO アプリケーションなどの一部のデータベースでは、通話明細記録（CDR）が極めて迅速に収集されます。規制目的のために、これらの記録を数年間保持することが必要とされており、その結果データベースのサイズが急増します。CDR テーブルがパーティション分割されている場合は、古いパーティションを低コスト階層に再配置できます。ただし、CDR テーブルがパーティション分割されていない場合は、代替手段が必要となります。代替手段の例を以下に示します。

CDR テーブルのエクステントは、要求に応じて割り当てられます。これは、エクステント 100 がエクステント 900 よりもかなり前に作成されることを意味します。また、エクステント 100 における CDR レコードの操作のほうが少なくなることを意味します。extent-id から、dba_extents および dba_segments 辞書テーブルを使って、ファイル、オフセット、長さの情報を生成できます。この情報から、ファイル内の特定された部分をセカンダリ階層に再配置できます。

低コスト階層またはセカンダリ階層に古いテーブルデータを再配置するには

1. Oracle AWR レポートを使って、データベースでアクティブでないオブジェクトまたはセグメントを特定します。
2. アクティブでないオブジェクトまたはセグメントを再配置できるストレージ階層を決定します。
3. SFDB リポジトリが最新の状態であることを確認するために、dbed_update を実行します。
4. テーブル *CDR_100* を *tier-2* に移動するコマンドを実行します。

```
$ dbdst_obj_move  -S PROD -H /home_dir -t   CDR_100 -c tier-2
```

以下に示す例のコマンドは、指定されたテーブルまたはインデックスが存在する一連のデータファイルを表示します。基になるファイルシステムが MVS ファイルシステムである場合は、対応するボリュームと階層名が表示されます。

データベースセグメントが正しい階層に再配置されていることを確認するには

◆ 再配置されたテーブルを表示するコマンドを実行します。

```
$ dbdst_obj_view  -S PROD -H /home_dir -o CDR_100
```

`dbdst_obj_move` コマンドの使用方法については、

p.283 の「SmartTier for Oracle を使用したサブファイルのデータベースオブジェクトの配置の最適化」を参照してください。

24

ストレージコストを最適化するためのファイルとデータベースの圧縮

この章では以下の項目について説明しています。

- 圧縮ファイルとデータベース

圧縮ファイルとデータベース

ファイルの圧縮は、データベース環境のストレージコストの削減をサポートします。Oracle データベースでは、アーカイブログ、パーティションに分割されたテーブル、低い頻度でアクセスされる表領域とデータファイルのストレージコストを削減する場合に、圧縮は優れた価値を提供します。データベースファイルの圧縮比は、データファイルに保存されるオブジェクトのタイプによって決まります。従来 Oracle は、TABLE と INDEX をデータファイルに保存しており、TABLE および INDEX に格納されるキーのタイプに関連付けられる列のタイプに応じて、圧縮比が検証されます。また、Oracle には、TABLE 内にある XML、表計算、MS Word 文書、写真など非構造型のデータを Secured Files 機能を使用して保存する機能もあります。これらのタイプの非構造型データは、圧縮に非常に適した候補です。アーカイブログは最大 90%、Oracle データファイルとインデックスは約 50% から 65% の圧縮を達成できます。

Oracle データベースファイルは、データベースが有効であれば、必要に応じて圧縮および圧縮解除できます。ただし、データベースのパフォーマンスに重大な影響を与えます。I/O 応答時間が減少するだけでなく、Oracle データベースがオンラインになっており、ファイルに対するトランザクションをアクティブに実行している間に、圧縮はシームレスに実行されます。圧縮は、ダイレクト I/O、非同期 I/O、同時 I/O、ODM、Cached ODM など、高度な I/O の方法でシームレスに動作します。データファイルに更新や新しい挿入があると、書き込みに関連する部分のファイルが圧縮解除されます。クエリーはメモリ内の圧縮されたデータを取得し、ファイルは圧縮解除された状態のままになります。

メモ: Quick I/O ファイルは圧縮できません。

DBA ユーザーとして `vxcompress` コマンドを実行できます。

次の使用例はデータベースに適用されます。

- 「サポートされるデータベースバージョンと環境」
- 「アーカイブログの圧縮」
- 「読み取り専用表領域の圧縮」
- 「アクセス頻度が低いテーブルパーティションの圧縮」
- 「アクセス頻度が低いデータファイルの圧縮」
- 「Oracle データベースのファイルの圧縮の推奨設定」

サポートされるデータベースバージョンと環境

SF (Storage Foundation) 、SFHA (Storage Foundation and High Availability) 、SF Oracle RAC (Storage Foundation for Oracle RAC) 、SFCFSHA (Storage Foundation Cluster File System High Availability) で圧縮がサポートされます。SF Oracle RAC、SFCFSHA などのクラスタ環境では、Veritas は負荷が最小のノードでファイルを圧縮することを推奨します。高速フェールオーバーの SFCFSHA 環境では、Veritas はデータベースがオフラインになっているパッシブノードでファイルを圧縮することを推奨します

アーカイブログの圧縮

アーカイブログは、データベースのリカバリに必要な重要ファイルです。ビジー状態の OLTP（オンライントランザクション処理）データベースでは、毎日アーカイブログが数 GB 生成されます。企業のガイドラインでは、アーカイブログを数日間保存するように指示することが多いです。Oracle のアーカイブログは読み取り専用ファイルで、生成された後に更新されることはありません。リカバリの間に、Oracle はアーカイブログを順番に読み取ります。したがって、アーカイブログは非常に圧縮に適した候補者です。また、アーカイブログは高圧縮性です。

次の例の手順では、1 日以上前のアーカイブログをすべて圧縮します。

1 日以上前のアーカイブログをすべて圧縮するには

1 Oracle DBA として次のクエリーを実行し、アーカイブログの場所を取得します。

```
SQL> select destination from v$archive_dest where status = 'VALID'
and valid_now = 'YES';
```

アーカイブログの送信先として /oraarch/MYDB を想定します。

2 1 日以上前のアーカイブログをすべて圧縮します。

```
$ find /oraarch/MYDB -mtime +1 -exec /opt/VRTS/bin/vxcompress {}
 ¥;
```

この手順は、cron のようなスケジューラを使用して毎日実行できます。

読み取り専用表領域の圧縮

大規模のデータベース環境では、読み取り専用のモードで変更がない静的な表領域を維持する一般的な方法です。読み取り専用表領域の主な目的は、大規模データベースの静的な部分のバックアップとリカバリを実行する必要性を取り除くことです。また、ユーザーが履歴データを変更できないように、履歴データを保護する方法も提供します。表領域を読み取り専用にすると、表領域に存在するすべてのテーブルとオブジェクトが、ユーザー更新の権限レベルに関係なく、更新されなくなります。これらのタイプの読み取り専用表領域は、圧縮に非常に適した候補です。月末レポートなどの例では、これらの読み取り専用表領域に対して実行される大きいクエリーがある場合があります。レポートをより迅速に実行するには、月次レポートを実行する前に、表領域をオンデマンドで圧縮解除できます。

次の例では、スポーツ用品の会社で、インベントリが winter_items と summer_items という 2 つの表領域に分割されています。春シーズンが終わると winter_item 表領域を圧縮し、summer_item 表領域を圧縮解除できます。夏シーズンの終わりには逆に処理できます。次の例の手順で、これらのタスクを実行します。

季節ごとに表領域を圧縮または圧縮解除するには

1 SQL を使用して、各表領域のファイルのリストを取得し、その結果を summer_files ファイルと winter_files ファイルに保存します。

```
SQL> select file_name from dba_data_files where
tablespace_name = 'WINTER_ITEM';
```

結果を winter_files ファイルに保存します。

```
SQL> select  file_name from dba_data_files where
tablespace_name = 'SUMMER_ITEM';
```

結果を summer_files ファイルに保存します。

2 winter_files ファイルを圧縮します。

```
$ /opt/VRTS/bin/vxcompress `/bin/cat winter_files`
```

3 summer_files ファイルを圧縮解除します。

```
$ /opt/VRTS/bin/vxcompress -u `/bin/cat summer_files`
```

アクセス頻度が低いテーブルパーティションの圧縮

テーブルのパーティション分割は、大規模の Oracle データベースで頻繁に使用される機能です。テーブルのパーティション分割は、パラレルクエリーを使用するトランザクションを並列化できるため、データベースのクエリーと更新の効率が向上します。また、データベースのメンテナンスを簡単にし、テーブルの可用性を向上させます。パーティションがダウンしている場合、テーブルの対応する部分のみがオフライン状態になり、テーブルの残りの部分がオンライン状態のままになります。通信環境では、一般的なのは「call_details」テーブルを月単位または四半期単位でパーティションに分割する方法です。パーティションの内容は、パーティションが古くなるにつれてあまり使用されなくなります。新しいパーティションに新しいレコードが追加され、以前の四半期のレコードは更新されません。通常は通信データベースは非常に大きいため、去年のデータを圧縮すると大幅に節約されます。

次の例では、テーブル「CALL_DETAIL」は四半期単位でパーティションに分割され、パーティションの名前は CALL_2010_Q1、CALL_2010_Q2、CALL_2011_Q1 などのように想定されます。2011 年の第 1 四半期では、CALL_2010_Q1 のデータを圧縮できます。

CALL_2010_Q1 パーティションを圧縮するには

1 CALL_2010_Q1 パーティションに属するファイル名を取得するには、SQL を使用します。

```
SQL> select tablespace_name from dba_tab_partitions
where table_name = 'CALL_DETAIL' and partition_name =
'CALL_2010_Q1';
```

クエリーが「TBS_2010_Q1」を返すことを想定します。

2 my_compress_files ファイルに名前を保存します。

```
SQL> select file_name from dba_data_files where
tablespace_name = 'TBS_2010_Q1';
```

結果を my_compress_files ファイルに保存します。

3 ファイルを圧縮します。

```
$ /opt/VRTS/bin/vxcompress `/bin/cat my_compress_files`
```

アクセス頻度が低いデータファイルの圧縮

多くの顧客データベースでは、Oracle のパーティション分割機能は使用されません。パーティション分割が使用されない場合、あまり使用されないデータファイルを特定するために Oracle カタログのクエリーを使用できます。カタログのテーブルを定期的にクエリーし、最も使用頻度の低いデータファイルを特定して、次の例の手順に示すようにそれらのファイルを圧縮します。

最も使用頻度の低いデータファイルを特定して圧縮するには

1 v$filestat をクエリーし、最も使用頻度の低いデータファイルを特定します。

```
SQL> select  name, phyrds + phywrts 'TOT_IO' from v$datafile d
and v$filestat f where d.file# = f.file# order by TOT_IO;
```

2 最も I/O 負荷が低いファイルをレポートから選択し、それらのファイルを圧縮します。

```
$ /opt/VRTS/bin/vxcompress file1 file2 file3 ...
```

3 定期的にクエリーを何度も実行して、圧縮ファイルの I/O 負荷が増加しないように確認します。I/O 負荷が増加する場合、ファイルを圧縮解除します。

```
$ /opt/VRTS/bin/vxcompress -u file1 file2 file3 ...
```

Oracle データベースのファイルの圧縮の推奨設定

ファイルが圧縮されているときにOracleデータベースがエラーなしで動作する場合でも、圧縮ファイルの I/O 増加は、データベースパフォーマンスを低下させます。Oracle データファイルの圧縮には次のガイドラインを使用します。

- データベース制御ファイルを圧縮しないでください。
- 一時表領域に属するファイルを圧縮しないでください。
- システム表領域と SYSAUX 表領域に属するファイルを圧縮しないでください。
- 圧縮ファイルの I/O 負荷を定期的に監視し、I/O 負荷が増加したらファイルを圧縮解除してください。

25

圧縮アドバイザツールの使用

この章では以下の項目について説明しています。

- 圧縮アドバイザツールについて
- 圧縮アドバイザを使った Oracle アーカイブログの圧縮
- 圧縮アドバイザを使った圧縮の候補データファイルの表示
- 圧縮アドバイザを使った Oracle データファイルの圧縮
- 圧縮アドバイザを使った圧縮データファイルの表示
- 圧縮アドバイザを使った Oracle データファイルの圧縮解除
- 圧縮アドバイザのコマンドリファレンス

圧縮アドバイザツールについて

Veritas File System（VxFS）は下位のアプリケーションに対して透過的なファイルを個別に圧縮できる `vxcompress` ユーティリティを提供します。圧縮ファイルを読み込むアプリケーションは、メモリのみで圧縮解除される圧縮解除データを自動的に受信し、ディスク上でデータの一部は圧縮された状態のままになります。圧縮ファイルに書き込むアプリケーションでは、ファイルの一部はディスク上で圧縮解除されます。

圧縮アドバイザは、Oracle の単一インスタンスと Oracle RAC 環境の Oracle データベースファイル用の拡張圧縮機能を提供します。圧縮アドバイザは次の目的で使用できます。

- データベースのパフォーマンスに影響することなく、ストレージの保存のために圧縮可能な Oracle のデータファイルを表示します。
- データファイルを圧縮/圧縮解除します。

- 指定した過去の日数で生成されたアーカイブログを圧縮します。
- 圧縮データファイルを表示します。

/opt/VRTS/bin ディレクトリに存在する Compression Advisor のコマンド sfae_comp_adm は、DBA ユーザーによって実行されます。

メモ: データベースで FileSnap 操作を実行する場合、データベースの圧縮機能を使うことはできません。

---

p.308 の「圧縮アドバイザのコマンドリファレンス」を参照してください。

p.304 の「圧縮アドバイザを使った Oracle アーカイブログの圧縮」を参照してください。

p.305 の「圧縮アドバイザを使った圧縮の候補データファイルの表示」を参照してください。

p.306 の「圧縮アドバイザを使った Oracle データファイルの圧縮」を参照してください。

p.307 の「圧縮アドバイザを使った圧縮データファイルの表示」を参照してください。

p.308 の「圧縮アドバイザを使った Oracle データファイルの圧縮解除」を参照してください。

圧縮アドバイザを使った Oracle アーカイブログの圧縮

Oracle のアーカイブログは読み取り専用ファイルで、生成された後に更新されることはありません。リカバリの間に、Oracle はアーカイブログを順番に読み取ります。アーカイブログは非常に圧縮に適した候補者です。また、アーカイブログは高圧縮性です。

圧縮アドバイザのコマンド sfae_comp_adm を使ってアーカイブログを圧縮できます。

圧縮アドバイザを使ってアーカイブログを圧縮するには

◆ 次のコマンドを使います。

```
$ sfae_comp_adm -o compress_archive_logs -n 2 -d /arch/sfaedb

Compressing file /arch/sfaedb/1_4_773425912.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_2_767364459.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_3_773425912.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_2_773425912.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_4_767364459.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_3_767364459.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_5_773425912.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_1_773425912.dbf resides on /arch/sfaedb
Compressing file /arch/sfaedb/1_6_773425912.dbf resides on /arch/sfaedb
```

このコマンドは 2 日以上前に生成され、`/arch/sfaedb` に属しているすべてのアーカイブログを圧縮します。

p.303 の「圧縮アドバイザツールについて」を参照してください。

p.308 の「圧縮アドバイザのコマンドリファレンス」を参照してください。

圧縮アドバイザを使った圧縮の候補データファイルの表示

圧縮アドバイザを使用してデータベースのパフォーマンスに影響することなく圧縮可能な Oracle データファイルを特定できます。この目的のため、圧縮アドバイザは Oracle Automatic Workload Repository（AWR）スナップショットを使用します。AWR スナップショットで最小の I/O 負荷を示すデータファイルの圧縮が推奨されます。

圧縮アドバイザを使って候補データファイルを表示するには

◆ 次のコマンドを使います。

```
$ sfae_comp_adm -o display -t tbs1

Tablespace    Filename                    Amount of     Amount of

                                          Data Read     Data
Written
                                          (bytes)       (bytes)
TBS1          /data/sfaedb/tbs1_2.dbf     302           344
TBS1          /data/sfaedb/tbs1_1.dbf     1246          6064
```

このコマンドは、表領域 TBS1 のデータファイルを表示します。出力テーブルは I/O 負荷の増加順で表示されるため、I/O 負荷が最小のデータファイルは上の行に表示されます。

p.303 の「圧縮アドバイザツールについて」を参照してください。

p.308 の「圧縮アドバイザのコマンドリファレンス」を参照してください。

圧縮アドバイザを使った Oracle データファイルの圧縮

圧縮アドバイザを使用すると、指定した Oracle データファイルを圧縮してストレージコストを節減することができます。圧縮可能なデータファイルを決定するには圧縮の候補データファイル情報を使用します。圧縮する単一のデータのファイル名を指定することも、圧縮するデータファイルのリストを含む -f オプションでファイル名を指定することもできます。SmartTier を使用している場合は、すべてのデータファイルが圧縮可能な階層を指定できます。

次のデータファイルは圧縮アドバイザで圧縮できません。これらのデータファイルを圧縮しようとすると、警告が表示されます。指定したデータファイルのセットが次のカテゴリの 1 つに属する場合、これらのデータファイルは圧縮されず、警告が生成されます。

- SYSTEM 表領域に属するデータファイル
- SYSAUX 表領域に属するデータファイル
- Oracle の一時ファイル
- アンドゥの表領域に属するデータファイル
- 指定の Oracle インスタンスのデータファイルではないファイル
- Veritas File System（VxFS）にないファイル

圧縮アドバイザを使ってデータファイルを圧縮するには

◆ 次のコマンドを使います。

```
$ sfae_comp_adm -o compress_datafiles -t ARCHIVE
Finding datafiles on tier ARCHIVE,
please be patient as this may take a while ...
Found on tier ARCHIVE, following datafiles:
/datavset/mytbs5.dbf
/datavset/mytbs4.dbf
/datavset/mytbs1.dbf
/datavset/mytbs2.dbf
/datavset/mytbs3.dbf

Compressing file /datavset/mytbs5.dbf resides on /datavset
Compressing file /datavset/mytbs4.dbf resides on /datavset
Compressing file /datavset/mytbs1.dbf resides on /datavset
Compressing file /datavset/mytbs2.dbf resides on /datavset
Compressing file /datavset/mytbs3.dbf resides on /datavset
```

このコマンドは ARCHIVE の階層にあるすべてのデータファイルを圧縮します。

p.303 の「圧縮アドバイザツールについて」を参照してください。

p.308 の「圧縮アドバイザのコマンドリファレンス」を参照してください。

圧縮アドバイザを使った圧縮データファイルの表示

圧縮アドバイザを使って Oracle の圧縮データファイルのリストを表示できます。

圧縮アドバイザを使って圧縮データファイルを表示するには

◆ 次のコマンドを使います。

```
$ sfae_comp_adm -o  display -c
Tablespace    Filename                  Filesize      Compression

                                                        Ratio (%)
TBS1          /data/sfaedb/tbs1_2.dbf    1.01 MB        99.51%
TBS1          /data/sfaedb/tbs1_1.dbf    112 KB         99.11%
USERS         /data/sfaedb/users01.dbf   5.01 MB        78.98%

Total storage saving 5.15 MB (92.54%)
```

このコマンドは圧縮データファイルを表示します。最終的な概略は、ストレージの節約スペースの量を表示します。

p.303 の「圧縮アドバイザツールについて」を参照してください。

p.308 の「圧縮アドバイザのコマンドリファレンス」を参照してください。

圧縮アドバイザを使った Oracle データファイルの圧縮解除

圧縮アドバイザを使って、指定の Oracle データファイルを圧縮解除できます。圧縮解除する単一のデータのファイル名を指定することも、圧縮解除するデータファイルのリストを含む -f オプションでファイル名を指定することもできます。SmartTier を使用している場合は、すべてのデータファイルが圧縮解除可能な階層を指定できます。

圧縮アドバイザを使ってデータファイルを圧縮解除するには

◆ 次のコマンドを使います。

```
$ sfae_comp_adm -o uncompress_datafiles -t ARCHIVE

Finding datafiles on tier ARCHIVE,
please be patient as this may take a while ...
Found on tier ARCHIVE, following datafiles:
/datavset/mytbs5.dbf
/datavset/mytbs4.dbf
/datavset/mytbs1.dbf
/datavset/mytbs2.dbf
/datavset/mytbs3.dbf

Uncompressing file /datavset/mytbs5.dbf resides on /datavset
Uncompressing file /datavset/mytbs4.dbf resides on /datavset
Uncompressing file /datavset/mytbs1.dbf resides on /datavset
Uncompressing file /datavset/mytbs2.dbf resides on /datavset
Uncompressing file /datavset/mytbs3.dbf resides on /datavset
```

このコマンドは ARCHIVE の階層にあるすべてのデータファイルを圧縮解除します。

p.303 の「圧縮アドバイザツールについて」を参照してください。

p.308 の「圧縮アドバイザのコマンドリファレンス」を参照してください。

圧縮アドバイザのコマンドリファレンス

sfae_comp_adm コマンド構文は次のように説明され、そのオプションは 表 25-1 にリストされます。

```
sfae_comp_adm [ -S ORACLE_SID ] [ -H ORACLE_HOME ]
-o compress_archive_logs [ -n N ] [ -d DESTINATION ]

sfae_comp_adm [ -S ORACLE_SID ] [ -H ORACLE_HOME ]
-o { compress_datafiles | uncompress_datafiles }
DATAFILE1 [ DATAFILE2 ... ]

sfae_comp_adm [ -S ORACLE_SID ] [ -H ORACLE_HOME ]
-o { compress_datafiles | uncompress_datafiles }
-f LISTFILE

sfae_comp_adm [ -S ORACLE_SID ] [ -H ORACLE_HOME ]
-o { compress_datafiles | uncompress_datafiles }
-t TIER

sfae_comp_adm [ -S ORACLE_SID ] [ -H ORACLE_HOME ]
-o display [ -n N ] [ -t TABLESPACE ] [ -c ]
```

表 25-1 圧縮アドバイザのコマンドオプション

オプション	用途
`-S` *`ORACLE_SID`*	使用する Oracle データベースの SID である ORACLE_SID を指定します。環境変数 *`ORACLE_SID`* を使用して環境にこの値を設定できます。
`-H` *`ORACLE_HOME`*	使用する Oracle インスタンスの HOME である ORACLE_HOME を指定します。環境変数 *`ORACLE_HOME`* を使用して環境にこの値を設定できます。
`-o compress_archive_logs`	指定したアーカイブログを圧縮します。
`-d` *`DESTINATION`*	`-o compress_archive_logs` とともに使用します。 指定した宛先から生成されたアーカイブログを圧縮します。データベースが複数の宛先へのアーカイブを設定されている場合、すべての宛先のアーカイブログはデフォルトで圧縮されます。
`-o compress_datafiles`	指定のデータファイルを圧縮します。
`-o uncompress_datafiles`	指定のデータファイルを圧縮解除します。

オプション	用途
-f *LISTFILE*	-o compress_datafiles または -o uncompress_datafiles とともに使用します。 圧縮または圧縮解除されるデータファイルの名前を含むファイルを各データファイルの名前 1 行で指定します。
-o display	圧縮可能なデータファイルの候補の一覧を表示します。
-t *TIER* -t *TABLESPACE*	■ -o compress_datafiles または -o uncompress_datafiles とともに使用する場合: すべてのデータファイルが圧縮または圧縮解除されるストレージ階層を指定します。 ■ -o display とともに使用する場合: 圧縮の候補であるデータファイルが表示される表領域を指定します。
-n *N*	■ -o compress_archive_logs とともに使用する場合: N 以上日前に生成されたアーカイブログを圧縮します。N に 0 を指定した場合、すべてのアーカイブログが圧縮されます。デフォルト値は 1 です。 ■ -o display とともに使用する場合: 表示できるファイルの数を指定します。デフォルト値は 10 です。
-c	-o display とともに使用します。 圧縮データファイルのリストを表示します。

p.303 の「圧縮アドバイザツールについて」を参照してください。

7

Oracle ディザスタリカバリの管理

- 第26章 レプリケート環境での Storage Foundation for Databases (SFDB) ツールの使用
- 第27章 レプリケート環境でのボリュームスナップショットの使用

26

レプリケート環境での Storage Foundation for Databases (SFDB) ツールの使用

この章では以下の項目について説明しています。

- SF Databases ツールと VVR（Volume Replicator）の配備について
- VVR（Volume Replicator）環境での Oracle データベースに対するデータベースアクセラレータの使用
- Oracle Data Guard での Storage Foundation for Databases（SFDB）ツールの配備について

SF Databases ツールと VVR（Volume Replicator）の配備について

Storage Foundation for Databases（SFDB）には、Storage Foundation Enterprise 製品の機能を強化するデータベース固有の拡張機能が備えられているため、Oracle データベースサーバーのパフォーマンス、可用性、管理性を向上させることができます。SFDB ツールでは、次の手法を使うことができます。

- AIX、Solaris の場合:
  データベースパフォーマンスを向上させる Quick I/O、Cached Quick I/O、ODM、Cached ODM などの拡張された I/O 方法
  Linux の場合:

データベースパフォーマンスを向上させる ODM、Cached ODM などの拡張された I/O 方法

- ファイルシステムチェックポイントとボリュームスナップショットの特定時点でのコピーの取得
- ボリュームスナップショットからのクローンデータベースの作成
- 任意のファイルシステムチェックポイントからのデータベースのロールバックの実行
- SmartTier for Oracle を使ったストレージ階層化の実行

Volume Replicator（VVR）は、VxVM（Veritas Volume Manager）の完全に統合されたコンポーネントとして動作する VxVM のオプションです。 VVR は、VxVM の堅牢さ、使いやすさおよび優れたパフォーマンスという利点を利用すると同時に、VxVM にレプリケーション機能を加えます。

VVR では次を実行できます。

- 既存の VxVM 構成のレプリケート
- アプリケーションを実行中の透過的な設定
- レプリケーション元のボリュームにアプリケーションが書き込んだデータを、距離に関係なく 1 つ以上のリモートの場所にレプリケート
- アプリケーションデータの一貫性のあるコピーをリモートの場所に提供

ソースの場所で災害が発生した場合、リモートの場所のアプリケーションデータのコピーを使い、リモートの場所でアプリケーションを再起動できます。この項では、リモート VVR サイトで SFDB ツールを使う方法について説明します。

Oracle データベース環境で VVR レプリケーションをセットアップおよび設定する方法について詳しくは、

『Veritas InfoScale 7.4.2 レプリケーション管理者ガイド』を参照してください。

VVR（Volume Replicator）環境での Oracle データベースに対するデータベースアクセラレータの使用

データベースアクセラレータは、データベースパフォーマンスを向上させるために使用される高速な IO 方式です。

- Oracle Disk Manager（ODM）
- CODM（Cached Oracle Disk Manager）
- Quick I/O（QIO）
- Cached Quick I/O（CQIO）

VVR（Volume Replicator）でレプリケートされたグローバル環境で SFDB ツールを使用する場合

- リモートサイトで Oracle のデータベースアクセラレータを使う場合は、SFDB リポジトリをレプリケートする必要がないため、リポジトリをレプリケートする必要はありません。
- プライマリサイトとリモートサイトで、同じ I/O 方式を使う必要はありません。ただし、リモートサイトでも同等のアプリケーションのパフォーマンスが必要な場合、リモートサイトでも同じ I/O 方式を使います。
- Quick I/O ファイルがレプリケートされているので、SFDB コマンドの qio_getdbfiles と qio_convertdbfiles を実行する必要はありません。

Oracle でデータベースアクセラレータを使うためのベストプラクティス

- プライマリサイトとリモートサイトで、必ず同じバージョンの Oracle を使用すること。
- リモートサイトで適切な ODM のライブラリをリンクして、必ず ODM を有効にすること。
- Quick I/O ファイルはリモートサイトの Quick I/O ファイルとしてレプリケートされるため、追加の設定は必要ありません。ただし、Cached Quick I/O を使っており、リモートサイトで Cached Quick I/O を使う場合は、`qioadmin` コマンドを使ってマウントプロパティを変更し、/etc/vx/qioadmin ファイルをプライマリからリモートにコピーする必要があります。

メモ: Quick I/O と Cached Quick I/O は Linux ではサポートされていません。

Oracle Data Guard での Storage Foundation for Databases（SFDB）ツールの配備について

SFDB ツールの機能は、Oracle Data Guard 環境のスタンバイデータベースをサポートするように拡張されています。

Oracle Data Guard の用語

- プライマリデータベース: このデータベースは「実稼動データベース」とも呼ばれ、ほとんどのユーザーアプリケーションによってアクセスされます。
- スタンバイデータベース: スタンバイデータベースは、トランザクションの整合性がとれたプライマリデータベースのコピーです。
- Data Guard 設定: Data Guard 設定は、1 つの実稼動データベースと 1 つ以上のスタンバイデータベースで構成されます。
- 物理スタンバイデータベース: ブロック単位でプライマリデータベースと同じオンディスクデータベース構造を持つ、プライマリデータベースと物理的に同一のコピーです。

次のポイントインタイムコピーの方法は、物理スタンバイデータベースで使うことができます。

- Database Storage Checkpoint
- サードミラーブレークオフスナップショット

Oracle Data Guard によるスタンバイ読み書きアクセスデータベースのクローン作成

- 物理スタンバイデータベースは読み取り専用データベースです。通常の状況では、物理スタンバイデータベースの内容を更新できないため、使用できるユーティリティ値は限られます。
- Data Guard サポート機能を使うと、DBI コマンドは読み取り専用スタンバイデータベースのクローンを作成し、書き込み可能なデータベースにすることができます。

Oracle Active Data Guard 機能が使われる環境では、データベーススナップショットを取得する前にスタンバイデータベース上で REDO ログの適用を明示的に停止する必要があります。

Active Data Guard 環境で Database FlashSnap を使うには

1. REDO ログの適用を停止します。

   次に例を示します。

   ```
   SQL>ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
   ```

2. dbed_update を実行します。
3. スナッププランを作成し、有効性を確認します。
4. スナップショットを取得します。
5. REDO ログの適用を開始します。

   次に例を示します。

   ```
   SQL>ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;
   ```

6. スナップショットのクローンを作成します。

27

レプリケート環境でのボリュームスナップショットの使用

この章では以下の項目について説明しています。

- ボリュームスナップショットを VVR と併用する場合の考慮事項
- VVR セットアップでの領域最適化スナップショットの作成
- VVR セットアップでのサードミラーブレークオフスナップショットの作成
- レプリケーションロールの転送後に vxsfadm を使用するための注意事項

ボリュームスナップショットを VVR と併用する場合の考慮事項

プライマリホストの Oracle データベースがセカンダリホストにレプリケートされる VVR（Volume Replicator）セットアップで領域最適化スナップショットまたはサードミラーブレークオフのスナップショットを使う予定がある場合、SFDB ツールに関する以下の注意事項が適用されます。

- Oracle 単一インスタンスデータベースは Storage Foundation スタックで実行され、すべてのボリュームが 1 つのディスクグループに含まれます。
- データとアーカイブログボリュームはセカンダリホストにレプリケートされます。
- すべてのアプリケーションボリュームは単一の RVG（Replicated Volume Group）に含まれます。
- VR セカンダリサイトのスナップショットまたはクローンについては、インスタントおよびオンラインスナップショットモードのみがサポートされます。インスタントモードスナップ

ショットは Oracle RAC データベースをサポートしません。そのため、インスタントモードのスナップショットを使う予定がある場合は、プライマリ VVR サイトは Oracle 単一インスタンス設定のみを使う必要があります。

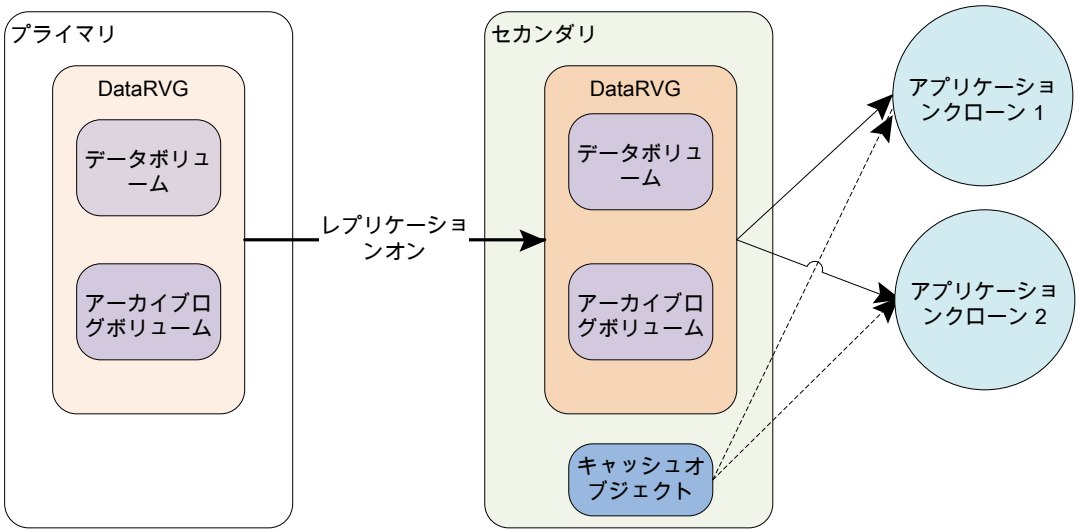
- 複数のセカンダリホストに対するレプリケーションはサポートされます。

VVR セットアップでの領域最適化スナップショットの作成

Volume Replicator（VVR）に基づいたレプリケーション環境で領域最適化スナップショットを使うには、プライマリホストでスナップショット設定を生成してから、その設定をセカンダリホストで使う必要があります。

図 27-1 には、領域最適化スナップショットを使うことのできるレプリケートされたデータベースの設定の概要が記載されています。

図 27-1　領域最適化スナップショット - レプリケートされたデータベースの設定



VVR セットアップで領域最適化スナップショットを作成するには

1 プライマリホストで、セカンダリホストのスナップショットの設定を生成します。

```
$ opt/VRTSdbed/bin/vxsfadm -a oracle -o getvvrinfo
Gathering offline data...                                Done

ORACLE_SID           = sfaedb
REPLICATION_ROLE     = secondary
REPLICATION_DG_VOL   = orasecondarydg,orasecondarydg2:datavol
SNAPSHOT_TIME        = Wed Feb 15 09:50:17 2012

Configuration file /tmp/config1 is ready.
Use this configuration file on replication
secondary site for cloning application instance.
```

メモ: 設定名とレプリケーションロールを指定する必要はありません。オンラインモードまたはインスタントモードのスナップショットでは、同じ設定ファイルを任意のセカンダリサイトで使うことができます。

---

2 新しく生成されたスナップショットの設定をプライマリホストからセカンダリホストにコピーします。

3 セカンダリホストで、スナップショットの設定を検証してから、スナップショットを作成します。

```
$ /opt/VRTS/bin/vxsfadm -s sos -a oracle -o valid -c config1
Validating database configuration for space-optimized snapshot:
ORACLE_SID               = sfaedb
ORACLE_HOME              = /ora_base/db_home
APP_MODE                 = online
Database validation successful.
Validating database volume layout for space-optimized snapshot:
REPLICATION_DG_VOL      = orasecondarydg,orasecondarydg2:datavol
CACHE_OBJECT             = cacheobj
Replicated data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup orasecondarydg.
SNAPSHOT_VOL_PREFIX      = SNAP_
Database volume layout validated successfully.
```

```
$ opt/VRTS/bin/vxsfadm -s sos -a oracle  -o snap -c config1
Creating snapshot volumes for data volumes ...
Done
```

セカンダリホストでスナップショット設定を検証した後で、セカンダリホストでクローンや破棄などの操作をローカルで実行できます。

4 セカンダリホストで、領域最適化スナップショットを使ってクローンを作成します。

```
$ /opt/VRTS/bin/vxsfadm -s sos -a oracle -o clone -c config1
Generated mount path: /var/tmp/az8EHWQpVe
Mounting snapshot volumes ...                          Done
Preparing parameter file for clone database ...        Done
Mounting clone database ...                            Done
Starting database recovery ...
Clone database sfaHrtyd is open.
```

5 セカンダリホストで、必要のなくなったクローンデータベースをマウント解除および破棄します。

```
$ /opt/VRTS/bin/vxsfadm -s sos -a oracle -o unmount -c config1
Shutting down clone database sfaHrtyd ...
Database shut down.
Unmounting snapshot volumes ...                          Done

$ /opt/VRTS/bin/vxsfadm -s sos -a oracle -o destroy -c config2
Destroying snapshot volumes ...                          Done
```

6 クローンデータベースを再起動します。

```
bash-2.05b$ /opt/VRTS/bin/vxsfadm -s sos -a oracle -c config1¥
-o clone --clone_name clones --clone_path /tmp/clones
Mounting snapshot volumes ...
Done
Starting clone database clones ...
Clone database started.
```

vxsfadm コマンドは、レプリケーションで役割を転送した後で使うことができます。

p.324 の「レプリケーションロールの転送後に vxsfadm を使用するための注意事項」を参照してください。

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.333 の「領域最適化スナップショットの設定パラメータ」を参照してください。

p.335 の「領域最適化スナップショットがサポートされる操作」を参照してください。

VVR セットアップでのサードミラーブレークオフスナップショットの作成

Volume Replicator（VVR）に基づいたレプリケーション環境でオンラインのサードミラーブレークオフ（FlashSnap）スナップショットを使うには、プライマリホストでスナップショット設定を生成してから、その設定をセカンダリホストで使う必要があります。

VVR セットアップでサードミラーブレークオフスナップショットを作成するには

1 プライマリホストで、セカンダリホストのスナップショットの設定を生成します。

```
$ opt/VRTSdbed/bin/vxsfadm -a oracle -o getvvrinfo
Gathering offline data...                        Done

ORACLE_SID           = sfaedb
REPLICATION_ROLE     = secondary
REPLICATION_DG_VOL   = orasecondarydg,orasecondarydg2:datavol
SNAPSHOT_TIME        = Wed Feb 15 09:50:17 2012

Configuration file /tmp/config.aTKYn is ready.
Use this configuration file on replication
secondary site for cloning application instance.
```

メモ: 設定名とレプリケーションロールを指定する必要はありません。同じ設定ファイルはオンラインモードまたは期間のモードのスナップショットの任意のセカンダリサイトで使うことができます。

---

2 新しく生成されたスナップショットの設定をプライマリホストからセカンダリホストにコピーします。

3 セカンダリホストで、スナップショットの設定を検証します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -a oracle ¥
--name flash3 -o valid -c config.aTKYn
Validating database configuration for third-mirror-break-off
snapshot:
ORACLE_SID                  = sfaedb
ORACLE_HOME                = /home/ora11gr2/app/ora11gr2/product

                              /11.2.0/dbhome_1
APP_MODE                    = online
SNAPSHOT_ARCHIVE_LOG        = yes
Validating database volume layout for space-optimized snapshot:
REPLICATION_DG_VOL          = orasecondarydg:datavol
CACHE_OBJECT                = co
Replicated volumes ready for snapshot:
Volume/volume-set datavol of diskgroup orasecondarydg.
Volume/volume-set archvol of diskgroup orasecondarydg.
SNAPSHOT_VOL_PREFIX         = SNAP_
SNAPSHOT_DG_PREFIX          = SNAP_
Database volume layout validated successfully.
```

セカンダリホストでスナップショット設定を検証した後で、セカンダリホストでクローンや破棄などの操作をローカルで実行できます。

4 セカンダリホストで、サードミラーブレークオフスナップショットを作成します。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -a oracle ¥
--name flash3 -o snap -c config.aTKYn
snapshot started at Wed Feb 15 14:02:34 2012.
Initiating snapshot from Primary site ...
Message from Primary:
Putting database in backup mode ...                          Done
Message from Primary:
Taking database out of backup mode ...                       Done
Flushing online redo logs ...                                Done
Sending IBC message to Secondary ...                         Done
Message from Host system1.example.com:
Creating snapshot volumes for data volumes ...               Done
Message from Host system1.example.com:
Receiving IBC message from Primary ...                       Done
Creating snapshot volume for archivelog volume ...           Done
Copying snapshot information to snapshot volume ...          Done
Creating snapshot diskgroups ...                             Done
Deporting snapshot diskgroups ...                            Done
SNAP_orasecondarydg
snapshot ended at Wed Feb 15 14:02:55 2012.
```

次のコマンドを使うとスナップショットボリュームを元のボリュームに再同期することができます。

```
$ /opt/VRTS/bin/vxsfadm -s flashsnap -a oracle ¥
--name flash3 -o resync -c config.aTKYn
resync started at Wed Feb 15 14:16:47 2012.
Importing snapshot diskgroups ...                            Done
Joining snapshot diskgroups to original diskgroups ...       Done
Reattaching snapshot volumes to original volumes ...         Done
resync ended at Wed Feb 15 14:16:58 2012.
```

メモ: ゴールデンイメージとして FlashSnap スナップショットを使用して作成された領域最適化スナップショットが存在する間は、FlashSnap スナップショットは再同期できません。

---

p.215 の「FlashSnap スナップショットを使った複数のクローン作成」を参照してください。

vxsfadm コマンドは、レプリケーションで役割を転送した後で使うことができます。

p.324 の「レプリケーションロールの転送後に vxsfadm を使用するための注意事項」を参照してください。

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

p.329 の「FlashSnap の設定パラメータ」を参照してください。

p.332 の「FlashSnap がサポートされる操作」を参照してください。

p.333 の「領域最適化スナップショットの設定パラメータ」を参照してください。

p.335 の「領域最適化スナップショットがサポートされる操作」を参照してください。

レプリケーションロールの転送後に vxsfadm を使用するための注意事項

vxsfadm コマンドは、プライマリロールをセカンダリサイトに移行またはフェールオーバーする場合に使用できます。ファイルシステムレベルのスナップショット、Storage Checkpoint、プライマリサイトに作成した FileSnap スナップショットは、レプリケーションプライマリロールをプライマリからセカンダリに転送した後で使用可能になります。ただし、ボリュームレベルのスナップショット（サードミラーブレークオフスナップショットと領域最適化スナップショット）は、新しいプライマリサイトでは使用できません。ロール転送後に vxsfadm コマンドを使用するには、プライマリサイト、レプリケーションサイト、セカンダリサイトで生成されたスナップショット設定を検証する必要があります。vxsfadmコマンドはまたプライマリのロールがオリジナルのプライマリホストに戻って転送されれば使うことができます。

プライマリロールの転送後に vxsfadm コマンドを使用するには、以下の要件を満たす必要があります。

- アプリケーションキー（ORACLE_SID など）が、レプリケーションロールの転送中または転送後に変更されていないこと。
- SFDB リポジトリが、レプリケーションロールの転送後に使用可能になっていること。リポジトリバックアップツールを使用して、リポジトリを使用可能にすることができます。

p.326 の「vxsfadm コマンドリファレンス」を参照してください。

8

Storage Foundation for Databases 管理リファレンス

- 第28章 Storage Foundation for Databases コマンドリファレンス
- 第29章 Storage Foundation for Databases のチューニング
- 第30章 SFDB ツールのトラブルシューティング
- 第31章 Oracle データベースの手動によるリカバリ
- 第32章 6.0 より前のリリースの Storage Foundation for Databases のコマンドリファレンス

28

Storage Foundation for Databases コマンドリファレンス

この章では以下の項目について説明しています。

- vxsfadm コマンドリファレンス
- FlashSnap リファレンス
- 領域最適化スナップショットリファレンス
- FileSnap リファレンス
- Database Storage Checkpoints のリファレンス

vxsfadm コマンドリファレンス

vxsfadm は、Oracle データベースに対して FlashSnap、FileSnap、領域最適化スナップショット、Database Storage Checkpoint など、ポイントインタイムコピーのさまざまな操作を実行するために使用できるコマンドラインユーティリティです。このユーティリティは、Storage Foundation の下位機能を使用して以下の操作を実行します。

メモ: 表領域の追加または破棄やデータファイルの追加または破棄など、データベースへの構造上の変更が進行中である場合、SFDB ツールはポイントインタイムコピーの作成をサポートしません。ただし、ポイントインタイムコピーが作成済みであれば、データベースの状態にかかわらずいつでもクローンを作成できます。

---

vxsfadm ユーティリティは、以下のような複数のデータベース設定でサポートされます。

- Oracle 単一インスタンス
- オフホストでの Oracle 単一インスタンス
- 高可用性設定（VCS クラスタ）での Oracle 単一インスタンス
- Storage Foundation Cluster File System High Availability での Oracle の単一インスタンス
- Oracle RAC と Storage Foundation for Oracle RAC
- Veritas Volume Replicator または Oracle Data Guard を使用したレプリケート設定の Oracle

vxsfadm コマンドの構文は次のとおりです。

```
vxsfadm -s <service_type> -a <application-name> -o <operation>
[ -c config-file ]
[ -r repository-host ]
[ service-specific parameters ]
[ application-specific parameters ]

vxsfadm -s <service-type> -a application-name> -o list
[ service-specific parameters ]
[ application-specific parameters ]

vxsfadm -s <service-type> -a application-name> -o setdefaults
[ service-specific parameters ]
[ application-specific parameters ]
```

vxsfadm パラメータは、次のとおりです。

`-s service_type`

PITC 操作のサービスタイプを指定します。

サポートされるサービスタイプは、以下のとおりです。

- flashsnap: ボリュームレベルスナップショット
- sos: 領域最適化ボリュームレベルスナップショット
- filesnap: ファイルレベルのスナップショット
- checkpoint: ファイルシステムのチェックポイント

`-o operation`	実行するポイントインタイムコピー操作を指定します。サポートされる操作はサービスタイプによって異なります。サービスタイプでサポートされる操作について詳しくは、それぞれの `vxsfadm` man ページを参照してください。 以下の操作は、すべてのサービスタイプに共通です。 ■ `-o list`: 指定したアプリケーションのすべての設定/サービスを一覧表示します。*service_type* を指定すると、指定した *service_type* のすべての既存の設定が表示されます。 ■ `-o setdefaults`: ファイルに保存して残りの操作に使用できるデフォルトの設定を生成します。
`-a application_name`	ポイントインタイムコピー操作を実行するアプリケーションを指定します。
`-c config_file`	service_type、application_name、operation のほかに、アプリケーションに固有のすべてのコマンドラインオプションを設定ファイルに指定できます。設定ファイルの情報は、PARAMETER_NAME=VALUE という形式で指定する必要があります。サービスにサポートされるパラメータについては、`vxsfadm` のマニュアルページを参照してください。
`-r repository-host`	オフホストの操作のために SFDB リポジトリのホストを指定します。
`service-specific parameters`	サービスでは、処理を行うために、これらのパラメータが必要です。最も重要なサービス固有のパラメータは、*--service*_name です。たとえば、サードミラーブレークオフスナップショット（flashsnap）を作成する場合、service_name パラメータは flashsnap_name で、このパラメータの重複のない識別子を `--flashsnap_name=dailysnap` のように指定できます。この識別子は、そのコピーに対して実行されるすべての操作に使用できます。 これらのパラメータは、設定ファイルに指定することも、環境にエクスポートすることもできます。 詳しくは、`vxsfadm` man ページを参照してください。

`application-specific parameters`

アプリケーションでは、処理を行うために、これらのパラメータが必要です。最も重要なアプリケーション固有のパラメータは、アプリケーションインスタンスを一意に識別するパラメータです。

たとえば、2 つの Oracle データベースをシステムで実行している場合、oracle_sid パラメータと oracle_home パラメータは一意にアプリケーションインスタンスを識別します。これらのパラメータは特定のアプリケーションインスタンスに対するすべてのポイントインタイムコピー操作に使用されます。

これらのパラメータは、設定ファイルに指定することも、環境にエクスポートすることもできます。

詳しくは、`vxsfadm` man ページを参照してください。

メモ: コマンドラインに指定した `vxsfadm` パラメータは、設定ファイルに指定したパラメータと環境にエクスポートしたパラメータを上書きします。設定ファイルに指定したパラメータは、環境にエクスポートしたパラメータを上書きします。

`vxsfadm` とエラーメッセージをトラブルシューティングに使用できます。

p.372 の「SFDB ツールのトラブルシューティングのためのリソース」を参照してください。

詳しくは、以下のマニュアルページを参照してください。

```
vxsfadm-flashsnap(1M), vxsfadm-checkpoint(1M)

vxsfadm-sos(1M), vxsfadm-filesnap(1M)
vxsfadm-oracle-flashsnap(1M), vxsfadm-oracle-sos(1M),
vxsfadm-oracle-checkpoint(1M), vxsfadm-oracle-filesnap(1M)
```

FlashSnap リファレンス

この項では、FlashSnap 設定パラメータとサポートされている操作について説明します。

FlashSnap の設定パラメータ

表 28-1は、設定ファイルで指定できる FlashSnap の設定パラメータのリストです。コマンドラインでパラメータを指定する場合は、表中のコマンドラインオプションのカラムを参照してください。

表 **28-1** FlashSnap - 設定パラメータ

パラメータ	説明	指定できる値	コマンドラインオプション
VERSION	設定ファイル形式のバージョン。	n.n 例: 7	NA
FLASHSNAP_NAME (*)	FlashSnap 設定の一意の識別子。	文字列 例: snap1	`--flashsnap_name snap1` または `--name snap1`
ORACLE_SID (*)	スナップショットを作成するデータベースの SID。	文字列 例: orcl	`--oracle_sid orcl` または `-S orcl`
ORACLE_HOME (*)	データベースの SID の Oracle ホームディレクトリ。	有効なパス 例: `/oracle/`*`databasehome`*	`--oracle_home /oracle/`*`databaseHome`* または `-H /oracle/`*`databaseHome`*
APP_MODE	スナップショット操作が実行されている場合のアプリケーションのモード。	offline online instant	`--app_mode offline` または `online` または `instant`
PFILE	標準以外の場所にある pfile を使用してデータベースを実行している場合のデータベース pfile の場所。	pfile の場所	--pfile の*場所*
SNAPSHOT_PLEX_TAG	スナップショットに含める必要があるプレックスの putil2 属性タグの値。	文字列 例: dbed_snap1	`--snapshot_plex_tag dbed_snap1` または `--plex_tag dbed_snap1`
SNAPSHOT_VOL_PREFIX	スナップショットボリューム名を作成するためにボリューム名の先頭に付加する文字列。	文字列 例: SNAPVOL_	`--snapshot_vol_prefix SNAPVOL_`
SNAPSHOT_DG_PREFIX	スナップショットディスクグループ名を作成するためにディスクグループ名の先頭に付加する文字列。	文字列 例: SNAPDG_	`--snapshot_dg_prefix SNAPDG_`
SNAPSHOT_MIRROR	スナップショットボリュームを形成するために切り離す必要があるミラーの番号。	番号	`--snapshot_mirror 2` または `--n 2`

パラメータ	説明	指定できる値	コマンドラインオプション
SNAPSHOT_ARCHIVE_LOG	このパラメータを設定する場合は、スナップショット操作をアーカイブログボリュームでも行う必要があります。 このパラメータは Oracle データベースにのみ適用されます。	Y または N	`--snapshot_archive_log` または `--no_snapshot_archive_log`
SECONDARY_HOST	スナップショットがマウントできアプリケーションのクローンを作成できるホスト。	ホスト名	`--secondary_host sys4`
PFILE_MOD	クローンインスタンスの初期化パラメータを変更するための Oracle pfile 形式のファイルの場所。	pfile の場所。	`--pfile_mod` *`pfile_location`*
MAPPED_MOUNTS	スナップショットボリュームをマウントするパス。	dg1:volume1=path1; dg2:volume2=path2 例: mydg:datavol=/tmp/fsnp;	`--mapped_mounts mydg:datavol=/tmp/fsnp`
CLONE_PATH	スナップショットボリュームをマウントするディレクトリ。	有効なパス `/tmp/so1`	`--clone_path/tmp/so1` または `mountpath/tmp/so1`
CLONE_NAME	クローンインスタンスの Oracle_SID。	文字列 例: clone1	`--clone_name clone1`
EXTRA_OBJECTS	スナップショットディスクグループに含む必要がある追加オブジェクトのリスト。追加オブジェクトの例: ボリューム、ボリュームセット、キャッシュオブジェクト。	各オブジェクトを <diskgroup>:<object> で指定した、オブジェクトのカンマ区切りリスト。	`--extra_objects` *`<dg1>:<vol1>,<dg2>:<vol2>`* または `--extra` *`<dg1>:<co1>`*
`REVERSE_RESYNC_ARCHIVE_LOG`	アーカイブログボリュームを逆再同期化するかどうか。	Yes または No (デフォルト: No)	`—reverse_resync_archive_log <value>` または `--rr_archive_log <value>`
`REVERSE_RESYNC_RECOVERY`	逆再同期化時のデータベースのリカバリ方法。	Auto、Manual、または Auto_Until_SCN (デフォルト: Auto)	`--reverse_resync_recovery <value>` または `--rr_recovery <value>`

メモ: PFILE_MOD パラメータで指定されるパスは、クローン操作が実行されるシステム対してローカルです。コマンド呼び出し元のホスト以外のホストに対してクローン操作を実行する場合は、クローン操作を実行するホストにパラメータファイルをコピーする必要があります。

メモ: (*) は、パラメータが必須であることを表します。

FlashSnap がサポートされる操作

「表 28-2」は、vxsfadm コマンドを使用して実行できる FlashSnap 操作の一覧です。

表 28-2 FlashSnap がサポートされる操作

操作	操作のエイリアス	説明
setdefaults	なし	アプリケーション用のデフォルトの設定ファイルを生成します。
valid	validate	設定ファイルおよび環境を検証します。
snap	snapshot、break	ミラーを切り離してスナップショットを作成し、ディスクグループを分割します。
clone	clonedb	アプリケーションのクローンを作成します。clone オプションを使って、マウント解除されたクローンを再起動できます。
mount	snapmount	ミラーボリュームをマウントします。
unmount	umount、snapunmount、snapumount	スナップショットをマウント解除し、クローンを作成する場合にクローンを停止します。
destroy	delete、remove	FlashSnap 設定をリポジトリから破棄します。
resync	なし	ミラーボリュームをプライマリボリュームと再同期します。

操作	操作のエイリアス	説明
reverse_resync_begin	rrbegin、rrstart、reverseresyncstart	スナップショットボリュームをマウントし、Oracle インスタンスを起動します。
reverse_resync_abort	rrabort、rrcancel、reverseresynccancel	逆再同期プレビューセッションを終了し、元のデータボリュームに戻ります。
reverse_resync_commit	rrcommit、rrfinish、reverseresyncfinish	reverse_resync_begin 操作でプレビューした逆再同期の変更をコミットします。
preclone	mountDB	データベースのクローンを作成するために必要なすべての操作を実行し、データベースをマウント状態で開きます。データベース回復を手動で実行する必要があります。
list	なし	利用可能なスナップショット設定をリストします。

領域最適化スナップショットリファレンス

この項では、FlashSnap 設定パラメータとサポートされている操作について説明します。

領域最適化スナップショットの設定パラメータ

表 28-3 に、設定ファイルで指定できる領域最適化スナップショットの設定パラメータが一覧表示されています。コマンドラインでパラメータを指定する場合は、表中のコマンドラインオプションのカラムを参照してください。

表 28-3　　領域最適化スナップショット - の設定パラメータ

パラメータ	説明	指定できる値	コマンドラインオプション
SOS_NAME (*)	スナップショット設定の一意の識別子。	文字列 例: sossnap	`--sos_name sossnap` または `--name sossnap`
VERSION	設定ファイル形式のバージョン。	n.n 例: 7	NA

パラメータ	説明	指定できる値	コマンドラインオプション
ORACLE_SID (*)	スナップショットを作成するデータベースの SID。	文字列 例: orcl	`--oracle_sid orcl` または `--S orcl`
ORACLE_HOME (*)	スナップショットを作成する SID の Oracle ホームディレクトリ。	有効なパス 例: `/oracle/databasehome`	`--oracle_home /oracle/databaseHome`
APP_MODE	スナップショット操作が実行されている場合のアプリケーションのモード。	offline online instant	`--app_mode offline` または `online` または `instant`
MAPPED_MOUNTS	ユーザーが指定するボリュームからマウントポイントへのマッピング。	dg1:volume1=path1; dg2:volume2=path2 例: mydg:datavol=/tmp/so;	`--mapped_mounts mydg:datavol=/tmp/so`
CLONE_NAME	クローン操作の間に作成されるアプリケーションインスタンスの名前。	文字列 例: clone1	`--clone_name= clone1`
CLONE_PATH	クローンのファイルがマウントされるパス。	有効なパス `/tmp/so1`	`--clone_path/tmp/so1` または `--mount_path/tmp/so1` または `--on/tmp/so1` または `--at/tmp/so1`
SNAPSHOT_VOL_PREFIX	ミラーボリュームの名前へのプレフィックス。	文字列 例: SNAPVOL_ デフォルト: SNAP_	`--snapshot_vol_prefix SNAPVOL_` または `--vol_prefix SNAPVOL_`
SNAPSHOT_ARCHIVE_LOG	このパラメータを設定する場合は、スナップショット操作をアーカイブログボリュームでも行う必要があります。 このパラメータは Oracle データベースにのみ適用されます。	Y または N デフォルト: Y	`--snapshot_archive_log` または `--no_snapshot_archive_log`
PFILE_MOD	クローンインスタンスの初期化パラメータを変更するための Oracle pfile 形式のファイルの場所。 このパラメータは Oracle データベースにのみ適用されます。	pfile の場所。	`--pfile_mod pfile_location`

パラメータ	説明	指定できる値	コマンドラインオプション
PFILE	標準以外の場所にある pfile を使用してデータベースを実行している場合のデータベース pfile の場所。	pfile の場所。	`--pfile` *`pfile_location`*
CACHE_OBJECT	ディスクグループおよび対応するキャッシュオブジェクトのカンマ区切りの値。	str1:str2,str3:str4,... 例: mydg1:C1,mydg2:c=C2,...	`--cache_object mydg1:C1,mydg2:c=C2`
SOURCE_SNAPSHOT	領域最適化スナップショットのゴールデンイメージとして使用する FlashSnap 設定の名前。	文字列 例: dailysnap	`--source_snapshot dailysnap` または `--source dailysnap`

メモ: PFILE_MOD パラメータで指定されるパスは、クローン操作が実行されるシステム対してローカルです。コマンド呼び出し元のホスト以外のホストに対してクローン操作を実行する場合は、クローン操作を実行するホストにパラメータファイルをコピーする必要があります。

メモ: (*) は、パラメータが必須であることを表します。

領域最適化スナップショットがサポートされる操作

「表 28-4」は、`vxsfadm` コマンドを使用して実行できる領域最適化スナップショットの操作の一覧です。

表 28-4 領域最適化スナップショットがサポートされる操作

操作	操作のエイリアス	説明
setdefaults	なし	アプリケーション用のデフォルトの設定ファイルを生成します。
valid	validate	データベース環境とシステムの設定を検証します。
snap	スナップショット	指定されたキャッシュオブジェクトを使用してボリュームのスナップショットを作成します。

操作	操作のエイリアス	説明
clone	clonedb	アプリケーションのクローンを作成します。この操作を使って、マウント解除されたクローンを再起動できます。
mount	snapmount	指定した場所にボリュームをマウントします。
umount	snapunmount、umount、snapumount	マウントされたスナップショットをマウント解除します。
preclone	mountdb	データベースをマウントの状態で開きます。 **メモ:** この操作は Oracle データベースに固有です。
destroy	delete、remove	スナップショットを破棄します。
getvvrinfo	なし	VVR セットアップでレプリケートされたデータベースのクローンを作成するためのリモートサイトで使うことができるスナップショット設定を生成します。

FileSnap リファレンス

この項では、FileSnap 設定パラメータとサポートされている操作について説明します。

FileSnap 設定パラメータ

表 28-5は、設定ファイルで指定できる FileSnap の設定パラメータのリストです。コマンドラインでパラメータを指定する場合は、表中のコマンドラインオプションのカラムを参照してください。

表 28-5　FileSnap - 設定パラメータ

パラメータ	説明	指定できる値	コマンドラインオプション
VERSION (*)	設定ファイル形式のバージョン。	n.n 例: 7	NA
FILESNAP_NAME (*)	スナップショット設定の一意の識別子。	文字列 例: filesnap1	`--filesnap_name filesnap1`
ORACLE_SID (*)	スナップショットを作成するデータベースの SID。	文字列 例: orcl	`--oracle_sid orcl`
ORACLE_HOME (*)	スナップショットを作成する SID の Oracle ホームディレクトリ。	有効なパス 例: `/oracle/databasehome`	`--oracle_home /oracle/databaseHome` または `--H /oracle/databaseHome`
CLONE_NAME	クローン操作の間に作成されるアプリケーションインスタンスの名前。	文字列 例: clone1	`--clone_name clone1`
PFILE_MOD	クローン操作中に使用できる変更済みパラメータを含む pfile の場所。	pfile の場所。	`--pfile_mod pfile_location`
PFILE	データベースが PFILE を使用している場合で、PFILE が *ORACLE_HOME*/dbs 以外の場所にある場合の Oracle PFILE へのパス。	pfile の場所。	`--pfile pfile_location`

メモ: PFILE_MOD パラメータで指定されるパスは、クローン操作が実行されるシステム対してローカルです。コマンド呼び出し元のホスト以外のホストに対してクローン操作を実行する場合は、クローン操作を実行するホストにパラメータファイルをコピーする必要があります。

メモ: (*) は、パラメータが必須であることを表します。

FileSnap がサポートされる操作

「表 28-6」は、`vxsfadm` コマンドを使用して実行できる FileSnap の操作の一覧です。

表 28-6　FileSnap がサポートされる操作

操作	操作のエイリアス	説明
setdefaults	なし	アプリケーション用のデフォルトの設定ファイルを生成します。
snap	スナップショット	アプリケーションによって使用されるボリュームのスナップショットを作成します。
destroysnap	なし	スナップショットを破棄します。
clone	clonedb	アプリケーションのクローンを作成します。
destroyclone	なし	アプリケーションのクローンを破棄します。
restore	なし	スナップショットが作成されたファイルをプライマリファイルにリストアします。
list	なし	スナップショットおよびクローンを状態とともに一覧表示します。

Database Storage Checkpoints のリファレンス

この項では、FlashSnap 設定パラメータとサポートされている操作について説明します。

Database Storage Checkpoint の設定パラメータ

表 28-7 は、設定ファイルに指定できる atabase Storage Checkpoint 設定パラメータを一覧表示します。コマンドラインでパラメータを指定する場合は、表中のコマンドラインオプションのカラムを参照してください。

表 28-7　Database Storage Checkpoint - 設定パラメータ

パラメータ	説明	指定できる値	コマンドラインオプション
VERSION	設定ファイル形式のバージョン。	n.n 例: 6.0	NA

パラメータ	説明	指定できる値	コマンドラインオプション
CHECKPOINT_NAME (*)	ストレージチェックポイントの一意の識別子。	文字列 例: ckpt1	`--checkpoint_name ckpt1` または `--checkpoint-name ckpt1` または `-name ckpt1`
ORACLE_SID (*)	ストレージチェックポイントを作成するデータベースの SID。	文字列 例: ora11gr2	`--oracle_sid ora11gr2` または `-S ora11gr2`
ORACLE_HOME (*)	ストレージチェックポイントを作成する SID の Oracle ホームディレクトリ。	文字列 例: /oracle_home	`--oracle_home /oracle_home` または `--oracle-home /oracle_home` または `-H /oracle_home`
APP_MODE	スナップショット操作が実行されている場合のアプリケーションのモード。	offline online instant デフォルト: online	`--app_mode offline` または `online` または `instant`
CLONE_NAME	クローン操作の間に作成されるアプリケーションインスタンスの名前。	文字列 例: clone1	`--clone_name clone1` または `--clone-name clone1`
CLONE_PATH	クローンのファイルがマウントされるパス。	有効なパス `/tmp/ckpt1`	`--clone_path/tmp/ckpt1` または `--clone-path/tmp/ckpt1` または `--mount-path/tmp/ckpt1` または `--mount_path/tmp/ckpt1`
SNAPSHOT_REMOVABLE	削除可能なストレージチェックポイントを作成するのか、削除不可のストレージチェックポイントを作成するのかを指定するパラメータ。	Y または N デフォルト: removable Y	`--snapshot_removable` または `--no_snapshot_removable--snapshot-removable` または `--no-snapshot-removable`
DATAFILE_ROLLBACK	ロールバックするデータファイルのカンマで区切った一覧。	データファイルのカンマで区切った一覧。	`--datafile_rollback` *list* または `--datafile-rollback` *list*

パラメータ	説明	指定できる値	コマンドラインオプション
DATAFILE_LIST_ROLLBACK	ロールバックするデータファイルの改行で区切った一覧を含むファイル。	データファイルの一覧を含むファイルへのパス。	`--datafile_list_rollback` *`path`* または `--datafile-list-rollback` *`path`*
TABLESPACE_ROLLBACK	ロールバックする表領域の名前。	表領域の名前	`--tablespace_rollback` *`name`* または `--tablespace-rollback` *`name`*
PFILE_MOD	クローンインスタンスの初期化パラメータを変更するための Oracle pfile 形式のファイルの場所。 このパラメータは Oracle データベースにのみ適用されます。	pfile の場所。	`--pfile_mod` *`pfile_location`* または `--pfile-mod` *`pfile_location`*
PFILE	標準以外の場所にある pfile を使用してデータベースを実行している場合のデータベース pfile の場所。	pfile の場所。	`--pfile` *`pfile_location`*

メモ: PFILE_MOD パラメータで指定されるパスは、クローン操作が実行されるシステム対してローカルです。コマンド呼び出し元のホスト以外のホストに対してクローン操作を実行する場合は、クローン操作を実行するホストにパラメータファイルをコピーする必要があります。

メモ: (*) は、パラメータが必須であることを表します。

Database Storage Checkpoint のサポートされる操作

「表 28-8」は、`vxsfadm` を使用して実行できる Database Storage Checkpoint の操作の一覧です。

表 **28-8** Database Storage Checkpoint のサポートされる操作

操作	操作のエイリアス	説明
setdefaults	なし	アプリケーション用のデフォルトの設定ファイルを生成します。
ckpt	create、checkpoint、fsckpt、snap、snapshot	アプリケーション用に Database Storage Checkpoint を作成します。
mount	snapmount、ckptmount	Database Storage Checkpoint を読み取り専用モードまたは読み書きモードで、指定したマウントポイントにマウントします。
mountrw	snapmountrw、ckptmountrw	Database Storage Checkpoint を読み書きモードでマウントします（元のストレージチェックポイントのシャドーチェックポイントが作成され、読み書きモードでマウントされます）。
unmount/umount	ckptumount、unmount、snapunmount、snapumount	Database Storage Checkpoint をマウント解除します。
delete	destroym、remove、ckptremove	Database Storage Checkpoint を削除します。
clone	ckptclone、clonedb	Database Storage Checkpoint のクローンを作成します。
restore	なし	Database Storage Checkpoint からデータファイルを復元します。
list	なし	Database Storage Checkpoint を一覧表示します。

操作	操作のエイリアス	説明
getappdata	なし	データベースがオンライン状態のときにデータベース情報を集めます。この情報はオフラインのチェックポイント処理で使用します。

29

Storage Foundation for Databases のチューニング

この章では以下の項目について説明しています。

- その他のマニュアル
- VxVM のチューニングについて
- VxFS のチューニングについて
- Oracle データベースのチューニングについて
- AIX の仮想メモリマネージャのチューニングについて
- Oracle 用の Solaris のチューニングについて

その他のマニュアル

この章で説明されているチューニングに関するヒントや情報とあわせて、次に示すようなより詳細な情報を扱った出版物を利用してください。

- Oracle パフォーマンスチューニングのヒントと技術（Osborne McGraw-Hill）
- Oracle のインストールマニュアル（Oracle 社）
- Oracle パフォーマンスチューニング（O'Reilly & Associates）
- Oracle パフォーマンスチューニングと最適化（Sams Publishing）
- Bullet-Proofing your Oracle Database with HACMP: A Guide to Implementing AIX Databases with HACMP (IBM Corporation)
- IBM の設定とパフォーマンス RedBooks（IBM Corporation）
- Storage Foundation 管理者ガイド、「VxVM のパフォーマンスの監視」の章

VxVM のチューニングについて

VxVM（Veritas Volume Manager）は、小規模なシステムから大規模なサーバーまで、大多数の設定に合わせてチューニングされます。比較的小規模なシステム（ディスクドライブが 100 台未満）では、チューニングは不要です。Veritas Volume Manager により、すべての設定パラメータについて適切なデフォルト値が採用されます。ただし、大規模なシステムの設定では、容量と処理効率という観点からこれらのパラメータの詳細なチューニングが必要になる場合があります。

VxVM のチューニングにはさまざまな機構が働いています。多くのパラメータは、AIX のシステム管理インターフェースツール（SMIT）を使ってチューニングできます。また、VxVM のコマンドラインインターフェースを使ってチューニングする必要のある値もあります。

VxVM のチューニングについて詳しくは『 Storage Foundation 管理者ガイド』を参照してください。

ボリュームの I/O 統計情報の取得について

データベースを単一のボリューム上の単一のファイルシステムに作成した場合、通常はボリュームの I/O 統計情報を監視する必要はありません。データベースを複数のボリューム上の複数のファイルシステムに作成した場合、または時間の経過とともにボリューム設定の変更を行っている場合は、データベースのボリュームの I/O 統計情報の監視が必要になることがあります。

`vxstat` コマンドを使うと、VxVM の制御下にあるボリューム、プレックス、サブディスク、ディスクの動作についての情報がアクセスされ、統計情報の概略が標準出力に表示されます。これらの統計情報の最初には、システムが起動されたときまたは最後にカウンタが 0 にリセットされたとき以降の VxVM のアクティビティが示されます。VxVM オブジェクト名が指定されていない場合は、設定データベース内のすべてのボリュームの統計情報が報告されます。統計情報を報告するオブジェクトがあるデータベースのディスクグループを指定するには、`-g` オプションを使います。

VxVM では次の I/O 統計情報が記録されます。

- 操作回数
- 転送ブロック数（一度に複数のブロックを転送することができます）
- 平均操作時間（VxVM インターフェース経由の合計時間が反映されており、他の統計プログラムとの比較には適しません）

VxVM は、論理 I/O について、直前の 3 つの情報を記録します。内容には各ボリュームの読み取り、書き込み、原子コピー、読み取りの検証、書き込みの検証、プレックスの読み取り、プレックスの書き込みが含まれています。VxVM では、読み取り障害、書き込み障害、修正された読み取り障害、修正された書き込み障害など、他の統計データも保持します。ボリュームの統計情報の表示に加えて、`vxstat` コマンドでは、ボリュームを形成

するコンポーネントの詳細な統計情報も表示できます。利用可能なオプションについて詳しくは、vxstat(1M) のマニュアルページを参照してください。

統計情報を 0 にリセットするには、-r オプションを使います。統計情報のリセットは、すべてのオブジェクトまたは指定したオブジェクトのみに実行できます。操作直前の状態にリセットすると、実行した操作による影響のみを測定できます。

次に、vxstat コマンドを使って生成された出力例を示します。

```
  OPERATIONS           BLOCKS         AVG TIME(ms)

TYP  NAME        READ   WRITE     READ     WRITE    READ     WRITE

vol  blop           0       0        0         0     0.0       0.0

vol  foobarvol      0       0        0         0     0.0       0.0

vol  rootvol    73017  181735   718528   1114227    26.8      27.9

vol  swapvol    13197   20252   105569    162009    25.8     397.0

vol  testvol        0       0        0         0     0.0       0.0
```

vxstat の出力を使用して動作が集中するボリュームを特定する方法と、これらのボリュームを認識して別のレイアウトに変更する、または移動する方法についての追加情報が利用できます。

RAID 5 ボリュームでは、この他のボリューム統計情報が使えます。

vxstat（1M）マニュアルページを参照してください。

『Symantec Storage Foundation 管理者ガイド』の「処理効率の監視とチューニング」の章の「処理効率の監視」の節を参照してください。

VxFS のチューニングについて

VxFS（Veritas File System）では、一連のチューニングオプションを使用してファイルシステムパフォーマンスを最適化し、さまざまなアプリケーションの作業負荷に対応できるようにします。VxFS には、一部の I/O 動作を制御する、一連の I/O チューニングパラメータが用意されています。これらの I/O パラメータにより、ファイルシステムをストライピングまたは RAID 5 ボリュームに対応させ、単体ディスクに比べて優れたパフォーマンスを発揮できます。通常、大容量ファイルにアクセスするデータストリーミングアプリケーションでファイルシステムのチューニングが最も活用されます。

Quick I/O を使用している場合、これらのチューニングオプションの大半は、データベースパフォーマンスにほとんど影響を与えません。ただし、Quick I/O を使用しているときにファイルシステムパフォーマンスのデータを収集し、その情報を使用してシステムリソースを最も効果的に使用できるようにシステム設定を調整することができます。

空き領域の監視の動作

通常、VxFS はファイルシステムの空き領域が 10% を超えると最高のパフォーマンスを実現します。これは、ファイルシステムの空き領域が 10% 以上であれば、断片化を解消でき、十分なエクステント領域を確保できるからです。df コマンドを定期的に使って、空き領域を監視することをお勧めします。空き領域のないファイルシステムは、ファイルシステムのパフォーマンスに悪影響を与えます。このため、空き領域のないファイルシステムの場合、一部のファイルを削除するか、または領域を拡張する必要があります。

fsadm_vxfs（1M）のマニュアルページを参照してください。

断片化の監視について

断片化により、パフォーマンスや可用性が低下します。このため、fsadm の断片化チェック機能や再編成機能を定期的に使うことをお勧めします。

断片化で問題が発生しないようにするための最も簡単な方法は、cron コマンドを使って定期的な断片化解消の実行のスケジュールを設定することです。

断片化を解消するスケジューリングは、週単位（ファイルシステムの使用頻度が高い場合）または月単位（ファイルシステムの頻度が低い場合）に設定する必要があります。エクステントの断片化は、fsadm か df -os コマンドを使って監視する必要があります。

断片化を解消するスケジューリングは、週単位（ファイルシステムの使用頻度が高い場合）または月単位（ファイルシステムの頻度が低い場合）に設定する必要があります。エクステントの断片化は、fsadm コマンドを使って監視する必要があります。

断片化の程度の判断に使用できる 3 つの要因を次に示します。

- 8 ブロック未満のエクステントの空き領域の割合
- 64 ブロック未満のエクステントの空き領域の割合
- 64 ブロック以上のエクステントの空き領域の割合

断片化が解消されているファイルシステムには、次の特性があります。

- 8 ブロック未満のエクステントの空き領域が 1% 未満
- 64 ブロック未満のエクステントの空き領域が 5% 未満
- 64 ブロック以上の空きエクステントとして利用可能な領域が全ファイルシステムサイズの 5% 以上

断片化が解消されていないファイルシステムには、次の特性が 1 つ以上あります。

- 8 ブロック未満のエクステントの空き領域が 5% 以上
- 64 ブロック未満のエクステントの空き領域が 50% 以上
- 64 ブロック以上の空きエクステントとして利用可能な領域が全ファイルシステムサイズの 5% 未満

fsadm の実行を適切な間隔でスケジュールし、エクステントを再構成する前後に fsadm のエクステントの断片化チェック機能を実行することで、最適なエクステントの再構成を実行するスケジュールを決定できます。

事前チェック結果は、再構成前の断片化の程度を示します。断片化の度合いが不適切なパーセンテージに近づいている場合は、fsadm の実行間隔を短くする必要があります。断片化の度合いが低い場合は、fsadm の実行間隔を長くできます。

VxFS I/O パラメータのチューニングの動作

VxFS には、一部の I/O 動作を制御する、一連の I/O チューニングパラメータが用意されています。これらの I/O パラメータにより、ファイルシステムをストライピングまたは RAID 5 ボリュームに対応させ、単体ディスクに比べて優れたパフォーマンスを発揮できます。通常、大容量ファイルにアクセスするデータストリーミングアプリケーションでファイルシステムのチューニングが最も活用されます。

VxFS と Veritas Volume Manager を併用する場合、ファイルシステムは VxVM にボリュームのジオメトリを問い合わせ、自動的に I/O パラメータを設定します。mkfs はファイルシステムを作成するときに VxVM に問い合わせを行い、ボリュームジオメトリに合わせて自動的にファイルシステムを作成します。mkfs のデフォルトのジオメトリ情報を使わない場合は、-o align=n オプションにより VxVM から取得したジオメトリ情報に置き換えることができます。さらに、ファイルシステムのマウント時には、mount コマンドからも VxVM に問い合わせが発行され、I/O パラメータを読み取ります。

デフォルトパラメータが使えない場合、または VxVM を使っていない場合は、/etc/vx/tunefstab ファイルを使って I/O パラメータ値を設定できます。mount コマンドは、/etc/vx/tunefstab ファイルを読み取り、ファイルシステムに指定されたパラメータを展開します。tunefstab ファイルの値は、VxVM から取得した値を上書きします。ファイルシステムがマウントされている場合は、vxtunefs コマンドにチューニングパラメータをコマンドラインから指定したり、/etc/vx/tunefstab ファイルから読み取らせることで、I/O パラメータを変更できます。

vxtunefs コマンドを使うと、I/O パラメータの現在の値を出力できます。

vxtunefs（1M）と tunefstab（4）の各マニュアルページを参照してください。

VxFS I/O チューニングパラメータについて

VxFS I/O チューニングパラメータは次のとおりです。

read_pref_io　　読み取り要求サイズ。ファイルシステムでは、このパラメータと read_nstream 値を組み合わせて、先読みするデータの量を決定します。デフォルト値は 64 K です。

write_pref_io	書き込み要求サイズ。ファイルシステムでは、このパラメータと write_nstream 値を組み合わせて、書き込み I/O 要求のフラッシュを実行するサイズを決定します。デフォルト値は 64 K です。
read_nstream	read_pref_io の並列に読み取り可能な数。ファイルシステムでは、read_pref_io で read_nstream を乗算した積から先読みサイズを決定します。read_nstream のデフォルト値は 1 です。
write_nstream	write_pref_io の並列に書き込み可能な数。ファイルシステムでは、この値と write_pref_io で write_nstream を乗算した積から書き込み時のフラッシュバックの時期を決定します。write_nstream のデフォルト値は 1 です。
default_indir_ size	VxFS では、ファイルの i ノードに、可変サイズの直接エクステントを最大 10 個まで格納できます。これらのエクステントをすべて使ってしまうと、ファイルは固定サイズの間接エクステントを使用する必要があります。サイズは、ファイルが初めて間接エクステントを使用するときに設定されます。間接エクステントのデフォルト値は 8 K です。間接エクステントサイズのエクステントを領域上に確保できない場合、ファイルシステムは、書き込みに失敗する可能性があるため、大容量間接エクステントを使うことなく、ENOSPC を戻します。大容量ファイルを数多く処理するファイルシステムでは、8 K の間接エクステントは小さすぎます。間接エクステントを必要とする大容量ファイルは、少数の大容量エクステントの代わりに多数の小容量エクステントを使う方が理想的です。このパラメータを使うことにより、デフォルトの間接エクステントサイズが増えるため、少数の大容量間接エクステントを使ったファイルを構成できます。 このチューニングを使う場合は注意が必要です。default_indir_size に過大な容量を指定すると、ファイルに間接エクステントの領域を確保できず、書き込みに失敗します。通常は、ファイルシステム上のファイルが少数で大容量になるほど、default_indir_size パラメータにも大きい値を設定します。このパラメータの値には、通常、read_pref_io パラメータの乗数値を指定します。 このチューニングは、バージョン 4 のディスクレイアウトには適用されません。

`discovered_direct_iosz`
ファイル I/O 要求が `discovered_direct_iosz` の値を超えている場合は、ディスカバードダイレクト I/O として処理します。ディスカバードダイレクト I/O は、ダイレクト I/O と同様に非バッファ I/O として処理されますが、ファイルの拡張またはブロックの領域確保時には、i ノードに対する同期的なコミットを必要としません。大容量 I/O 要求では、ディスク I/O に比べて、ページキャッシュへのデータのコピーに要する CPU 時間が増えるばかりか、I/O データをバッファリングするメモリ領域が大きくなります。これらの I/O 要求には、ディスカバードダイレクト I/O を使うと、通常の I/O より効率よく処理できます。このパラメータのデフォルト値は 256 K です。

`initial_extent_size`
デフォルトの初期エクステントサイズを変更します。VxFS では、新しいファイルの 1 回目の書き込みを基に、ファイルに割り当てる最初のエクステントサイズを決定します。通常、最初のエクステントは 1 回目の書き込みサイズよりも大きい最小二乗値に設定されます。また、2 乗値が 8 K 未満であれば、最初のエクステント領域を 8 K に設定します。最初のエクステント割り当て後、ファイルシステムは、引き続き同じファイルにエクステントを割り当てるたびにエクステントサイズを増やします（`max_seqio_extent_size` を参照）。アプリケーションの多くでは、始め、8 KB 以下のバッファサイズを割り当ててファイルへの書き込みを実行するため、エクステントサイズの増加は、最初のエクステントを 2 倍にすることから始まります。`initial_extent_size` はデフォルトの初期エクステントサイズを大きい値に変更できるため、最小二乗値使用のポリシーはより大きい値から始まり、ファイルの初期状態から複数の小容量エクステントを割り当てることを回避します。このパラメータは、平均的に大容量ファイルを処理するファイルシステムでのみ指定します。結果として、このようなファイルシステムでは、ファイルごとのエクステント数が減り、断片化の発生を防ぐことができます。`initial_extent_size` は、ファイルシステムブロック単位で計算されます。

`max_direct_iosz`
ファイルシステムで発行するダイレクト I/O 要求の最大サイズ。より大きいサイズの I/O 要求を受け取った場合、`max_direct_iosz` のサイズのチャンクに分割されます。このパラメータに、I/O 要求へ一度に割り当てるメモリのサイズを指定します。全メモリの 20% を超える値は指定できません。

`max_diskq`
1 つのファイルあたりに生成されるディスクキューの最大値を制限します。ファイルシステムがファイルのデータをフラッシュするときに、フラッシュされるページ数が `max_diskq` を超えた場合、フラッシュ中のデータ量が減少するまでプロセスはブロックされます。実際のディスクキューは制限しませんが、フラッシュ処理によってシステムが応答しなくなることを防ぎます。デフォルト値は 1MB です。

`max_seqio_extent_size`
エクステントの最大値を増減します。ファイルシステムでは、ファイルへの順次書き込みに使うデフォルトの割り当てポリシーに準じるとき、ファイルへの 1 回目の書き込みに十分な領域が確保されている初期エクステントを割り当てます。追加のエクステントの領域確保には、確保する領域が規則的に増大するため（アルゴリズムでは新しいエクステントごとにファイルサイズを 2 倍にします）、各エクステントに対しより多くのデータ書き込みが可能になります。これにより、順次書き込みの継続を見込んでエクステントの総数を削減できます。ファイルへの書き込み終了時に未使用領域を他のファイルに解放します。この割り当てにより通常、エクステントサイズの増加は 2048 ブロックで止まり、1 つのファイルが未使用領域を不必要に確保しないようにします。
`max_seqio_extent_size` はファイルシステムブロックの単位で測定されます。

`qio_cache_enable`
Quick I/O ファイルのキャッシュを有効または無効にします。デフォルトの動作では、キャッシュは無効になります。キャッシュを有効にするには、`qio_cache_enable` を 1 に設定します。大容量メモリを備えたシステムでも、データベースがすべてのメモリをキャッシュとして常に使えるとはかぎりません。ファイルシステムのキャッシュをセカンドレベルのキャッシュとして有効にすることにより、パフォーマンスが向上する可能性があります。たとえば、データベースでテーブル内順次スキャンを実行する場合、ファイルシステムのキャッシュを有効にすることで、ファイルシステムはファイルの積極的な先読みが可能になり、その結果、検索速度が上がることが考えられます。

write_throttle

警告: write_throttle パラメータは、コンピュータシステムに大量のメモリが実装されているものの、ストレージデバイスは低速であるという特殊な状況において有効です。この設定では、fsync() などの同期操作の完了に時間がかかりすぎて、システムがハングアップしているように見えることがあります。これが発生する理由は、ファイルシステムがシステムパフォーマンスを低下させることなくディスクへ非同期にフラッシュする以上の速度で、ダーティページ(メモリのみの更新)を作成するためです。

write_throttle の値を低くすると、ディスクにフラッシュされるまでにファイルシステムによって生成されるファイル単位のダーティページ数が制限されます。ファイルのダーティページ数が write_throttle しきい値に達すると、ファイルシステムは、空き領域の有無にかかわらず、ディスクへのフラッシュを開始します。write_throttle のデフォルト値では、通常、多数のダーティページが生成されますが、高速なユーザー書き込みは維持されます。ストレージデバイスの速度によっては、write_throttle を低くすると、ユーザー書き込みのパフォーマンスは低下する可能性がありますが、ダーティページの数が制限されるため、同期化操作は格段に短い時間で処理できるようになります。

write_throttle 値を低く設定することによって書き込み要求が遅延することがあるため(たとえば、write_throttle 値を低く設定することにより、max_diskq の値までファイルディスクキューの数が増える可能性があり、そのため、ディスクキューが少なくなるまで書き込み要求の遅延が発生することがある)、システムに大容量の物理メモリと低速ストレージデバイスの双方が搭載されていないかぎり、write_throttle の値を変更しないことをお勧めします。

ファイルシステムを VxVM とともに使う場合、ボリュームジオメトリを基にした VxFS I/O チューニングパラメータはデフォルトを指定することをお勧めします。

ファイルシステムを VxVM 以外のハードウェアディスクアレイや論理ボリューム管理ツールと連動させる場合、論理ディスクジオメトリからパラメータを調整しようとします。ストライピングや RAID 5 の場合、通常 read_pref_io にストライプユニットのサイズを、read_nstream にストライプの並列アクセス数を指定します。ストライプアレイの場合は、write_pref_io と write_nstream に同数を、RAID 5 アレイの場合は、write_pref_io に全ストライプサイズ(パリティストライプを除いたすべてのストライプユニットの合計)を、write_nstream に 1 をそれぞれ指定します。

ディスク I/O を効率的に処理するアプリケーションについては、read_nstream で read_pref_io を乗算した積から読み取り要求を発行する必要があります。通常、read_nstream で read_pref_io を乗算する場合は、どの累数を使ってもパフォーマンスには十分なサイズとなります。書き込みの場合にも、この方法で write_pref_io と

write_nstream のパラメータを指定することにより、パフォーマンスを向上させることができます。ファイルシステムのチューニングには、実際の作業負荷を課してチューニングパラメータを指定することをお勧めします。

大容量ファイルに対して順次 I/O を実行するアプリケーションでは、discovered_direct_iosz を超えるサイズの I/O 要求を実行します。これにより、I/O 要求がディスカバードダイレクト I/O 要求として認識され、ダイレクト I/O と同様に非バッファ I/O として処理されますが、ファイルの拡張時に i ノードの更新に対する同期化処理を実行しません。キャッシュを超えるサイズのファイルの場合、非バッファ I/O の処理を指定することにより、有益なデータをキャッシュから溢れさせることなく、CPU オーバーヘッドを軽減できます。

Quick I/O インターフェースを使用したファイル I/O の統計情報の取得について

qiostat コマンドを使うと、VxFS ファイルシステム上の Quick I/O ファイルのアクティビティ情報にアクセス可能になります。このコマンドは、Quick I/O インターフェースを使用してファイルを最初に開いた時間からの、ファイルのアクティビティレベルの統計情報を報告します。収集した qiostat の統計情報は、最後に開かれた Quick I/O ファイルへの参照が閉じられるとリセットされます。

qiostat コマンドは次の I/O 統計を表示します。

- 読み取りと書き込みの操作の数
- 転送されるデータブロック(セクタ)の数
- 読み取りと書き込みの操作にかかる平均時間

Cached Quick I/O が使用されると、-l(長い形式)オプションが選択されている場合は qiostat はキャッシング統計情報も表示します。

qiostat 出力の例を次に示します。

```
                    OPERATIONS        FILE BLOCKS             AVG
TIME(ms)

 FILENAME          READ     WRITE    READ     WRITE      READ
WRITE

 /db01/file1         0        00       0       0.0       0.0

 /db01/file2         0        00       0       0.0       0.0

 /db01/file3     73017    181735  718528   1114227      26.8
27.9
```

```
 /db01/file4     13197     20252  105569    162009     25.8
397.0

 /db01/file5         0        00       0       0.0      0.0
```

利用可能なオプションについて詳しくは、qiostat(1M)のマニュアルページを参照してください。

I/O 統計データについて

ファイル I/O のパフォーマンスデータを収集したら、それを使用して、システムリソースを最も効率よく使用するためにシステム設定を調整することができます。

考慮するべき重要な統計が 3 つあります。

- ファイル I/O のアクティビティ
- ボリューム I/O のアクティビティ
- RAW ディスク I/O のアクティビティ

データベースがストライプボリュームで 1 つのファイルシステムを使用している場合、ファイル I/O アクティビティの統計情報のみに注意を払う必要があります。複数のファイルシステムがある場合、ボリューム I/O アクティビティも監視する必要がある場合があります。

最初に、既存の統計情報をすべて消去するために qiostat -r コマンドを実行します。統計情報を消去したら、通常のデータベースの作業期間の間は、データベースをしばらく稼働させます。たとえば多くのユーザーが存在するデータベースを監視する場合、収集された I/O 統計情報を表示する前に、統計が収集されるまで主な稼働期間中に数時間待機します。

アクティブなファイル I/O の統計情報を表示するには、qiostat コマンドを実行して、統計をしばらく表示する間隔(-i を使用)を指定します。このコマンドは、次のような統計情報のリストを表示します。

```
                    OPERATIONS       FILE BLOCKS        AVG TIME(ms)

 FILENAME         READ     WRITE    READ     WRITE      READ
WRITE

 /db01/cust1       218        36     872       144      22.8
55.6

 /db01/hist1         0        10       4       0.0      10.0

 /db01/nord1        10        14      40        56      21.0
75.0
```

```
 /db01/ord1          19         16       76         64       17.4
56.2

 /db01/ordl1        189         41      756        164       21.1
50.0

 /db01/roll1          0         50        0        200        0.0
49.0

 /db01/stk1        1614        238     6456        952       19.3
46.5

 /db01/sys1           0         00        0        0.0        0.0

 /db01/temp1          0         00        0        0.0        0.0

 /db01/ware1          3         14       12         56       23.3
44.3

 /logs/log1           0         00        0        0.0        0.0

 /logs/log2           0        217 0   2255        0.0        6.8
```

ファイル I/O 統計情報を使用して、操作数が異常に多いファイルや読み取りおよび書き込み時間が過度のファイルを特定できます。これが発生したら、「ホット」ファイルやビジー状態のファイルシステムを別のディスクに移動したり、I/O の負荷を分散するためにレイアウトを変更します。

```
Mon May 11 16:21:20 2015

/db/dbfile01                813      0       813        0      0.3   0.0

/db/dbfile02                  0    813         0      813      0.0   5.5

Mon May 11 16:21:25 2015

/db/dbfile01                816      0       816        0      0.3    0.0

/db/dbfile02                  0    816         0              816    0.0
  5.3

Mon May 11 16:21:30 2015

/db/dbfile01                  0      0         0                0    0.0
  0.0

/db/dbfile02                  0      0         0                0    0.0
  0.0
```

Veritas extension for Oracle Disk Manager を使用したファイル I/O 統計の取得

odmstat コマンドにより VxFS システム上の Oracle Disk Manager ファイルのアクティビティに関する情報を表示できます。詳しくは、odmstat（1M）のマニュアルページを参照してください。このコマンドでは、Oracle Disk Manager インターフェースでファイルを開いた時点からのアクティビティの統計情報を報告します。このコマンドには統計をゼロにするためのオプションがあります。ファイルが閉じられるとき、統計情報は破棄されます。

odmstat コマンドは次の I/O の統計を表示します。

- 読み取りと書き込みの操作の数
- 読み取りおよび書き込みが行われたデータブロック数
- 読み取りと書き込みの操作にかかる平均時間

I/O 統計を取得するには

◆ 次のようにして、odmstat コマンドを使用します。

```
# /opt/VRTS/bin/odmstat -i 5 /mnt/odmfile*
```

次に例を示します。

```
# /opt/VRTS/bin/odmstat -i 5 /db/odmfile*

                          OPERATIONS       FILE BLOCKS     AVG TIME(ms)

FILE NAME                READ     WRITE   READ     WRITE   READ   WRITE

Mon May 11 16:21:10 2015

/db/cust.dbf                0         0      0         0    0.0     0.0

/db/system.dbf              0         0      0         0    0.0     0.0

Mon May 11 16:21:15 2015

/db/cust.dbf              371         0    371         0    0.2     0.0

/db/system.dbf              0       371      0       371    0.0     5.7

Mon May 11 16:21:20 2015

/db/cust.dbf              813         0    813         0    0.3     0.0

/db/system.dbf              0       813      0       813    0.0     5.5

Mon May 11 16:21:25 2015

/db/cust.dbf              816         0    816         0    0.3     0.0

/db/system.dbf              0       816      0       816    0.0     5.3

Mon May 11 16:21:30 2015

/db/cust.dbf                0         0      0         0    0.0     0.0

/db/system.dbf              0         0      0         0    0.0     0.0
```

I/O 統計について

ファイルシステムを通してデータベースを実行するとき、ファイルごとに読み書き両用のロックを取得すると、1 ファイルにつき 1 つのアクティブな書き込みのみが可能となります。iostat を使用してディスクの統計を調べる場合、ディスクではキューに格納する時間お

よびサービス時間を報告します。サービス時間とはディスクで I/O が費やす時間であり、キューに格納する時間とはその I/O が先行する他のすべての I/O を待機する時間です。ボリュームレベルまたはシステムレベルではキューに格納することがないため、vxstat および qiostat ではキューに格納する時間が表示されません。

たとえば、100 個の I/O を同時に送信し、1 個につき 10 ミリ秒かかる場合、ディスクではサービス時間の 10 ミリ秒とキューに格納する時間の 490 ミリ秒の平均を報告します。vxstat、odmstat、qiostat は、サービス時間 500 ミリ秒の平均を報告します。

Oracle データベースのチューニングについて

Oracle データベースの最適なパフォーマンスを実現するには、VxFS とうまく連携するようにデータベースをチューニングする必要があります。Oracle データベースのパフォーマンスを向上するためにチューニング可能ないくつかの Oracle パラメータがあります。

順次テーブルスキャン

Quick I/O のデフォルトモードでは、すべての I/O をダイレクト I/O として実行します。単一スレッドの順次スキャンの場合（意思決定支援システム（DSS）の作業負荷で一般的）には、バッファ処理された読み取りを使用するとパフォーマンスの向上をもたらすことができます。ファイルシステムがこれらの順次読み取りを検出し先読みを実行するため、Oracle が要求する次の数ブロックがシステムバッファキャッシュですぐに利用できる状態になり、単に Oracle のシステムグローバル領域（SGA）にコピーされます。メモリからのアクセスはディスクからのアクセスよりも本質的に速いため、応答時間が大幅に短縮されます。

Quick I/O 使用時に大規模な順次スキャンを処理するには、次のいずれかの方法でパフォーマンスを向上させることができます。

- Oracle のパラレルクエリーオプションを使用して単一の大規模なスキャンを複数の小規模なスキャンに分割します。

  メモ: ご使用のシステムとバージョンに対応した Oracle のマニュアルを参照し、これらのパラメータに推奨される設定が提示されている場合はそれを使用してください。

- 2 つ目の方法は、Oracle の順次スキャンのプロセスによって読み込まれるファイルに対して Cached Quick I/O を有効にするというものです。Cached Quick I/O によりバッファ処理された読み取りが可能になり、この自動的なファイルシステムの先読みによってデータがプレロードされることで応答時間が短縮されます。

  メモ: Oracle の 64 ビット版を使用している場合は、このオプションを使わないでください。

順次テーブルスキャン

Quick I/O は、すべての I/O をダイレクト I/O として実行します。単一スレッドの順次スキャンの場合 (意思決定支援システム（DSS）の作業負荷で一般的) には、バッファ処理された読み取りを使用するとパフォーマンスの向上をもたらすことができます。ファイルシステムがこれらの順次読み取りを検出し先読みを実行するため、Oracle が要求する次の数ブロックがシステムバッファキャッシュですぐに利用できる状態になり、単に Oracle のシステムグローバル領域 (SGA) にコピーされます。メモリからのアクセスはディスクからのアクセスよりも本質的に速いため、応答時間が大幅に短縮されます。

Quick I/O 使用時に大規模な順次スキャンを処理するには、次のいずれかの方法でパフォーマンスを向上させることができます。

- Oracle の並列クエリーオプションを使用して単一の大規模なスキャンを複数の小規模なスキャンに分割します。

  メモ: ご使用のシステムとバージョンに対応した Oracle のマニュアルを参照し、これらのパラメータに推奨される設定が提示されている場合はそれを使用してください。

- 2 つ目の方法は Oracle8 を使用している場合のみ適用できます。この方法は、Oracle の順次スキャンのプロセスによって読み込まれるファイルに対して Cached Quick I/O を有効にするというものです。Cached Quick I/O によりバッファ処理された読み取りが可能になり、この自動的なファイルシステムの先読みによってデータがプレロードされることで応答時間が短縮されます。

非同期 I/O

Quick I/O および Oracle Disk Manager はカーネル非同期 I/O をサポートしており、これにより CPU 利用率が低減しトランザクションのスループットが向上します。

Oracle Disk Manager はカーネル非同期 I/O をサポートしており、これにより CPU 利用率が低減しトランザクションのスループットが向上します。

次のパラメータを有効にすると Oracle は非同期 I/O を利用できるようになり、DBWR の複数のスレーブを設定する必要がなくなります。

- Oracle9 で Quick I/O のデータファイルを使用している場合は、`init.ora` で `DISK_ASYNCH_IO` を `TRUE` に設定します。
- Oracle9 で ODM を使用している場合は、いずれの `init.ora` パラメータも変更する必要はありません。

Oracle のインストールガイドには、システムに非同期 I/O を実装する詳しい手順が説明されています。

バッファキャッシュのチューニング

JFS をバッファ付き I/O モードで使用する場合に、UNIX バッファキャッシュはパフォーマンスの点で重要な役目を果たします。

UNIX バッファキャッシュは UFS をバッファ付き I/O モードで使用する場合に重要な役目を担います。

ただし、Quick I/O 使用時には RAW デバイスを使用する場合と同じようにデータベースをチューニングする必要があります。Quick I/O ではデータベースのパフォーマンスが向上するようにファイルシステムのキャッシュをバイパスするため、データベースのバッファキャッシュにより多くのメモリを割り当てることができます。ファイルシステムのキャッシュに通常割り当てられるメモリページは、データベースのバッファキャッシュ（SGA）に割り当てることができます。データベースをシャットダウンせずに SGA サイズを調整できます。

バッファキャッシュのチューニング

UFS をバッファ I/O モードで使用する場合に、UNIX バッファキャッシュはパフォーマンスの点で重要な役目を果たします。

ただし、Quick I/O 使用時には RAW デバイスを使用する場合と同じようにデータベースキャッシュをチューニングする必要があります。Quick I/O ではデータベースのパフォーマンスが向上するようにファイルシステムのキャッシュをバイパスするため、データベースのバッファキャッシュにより多くのメモリを割り当てることができます。ファイルシステムのキャッシュに通常割り当てられるメモリページは、データベースのバッファキャッシュ (SGA) に割り当てることができます。Oracle9i を使うと、データベースを終了しないで SGA のサイズを調整できます。

順次スキャン中の Oracle のブロック読み取りの設定

`DB_FILE_MULTIBLOCK_READ_COUNT` パラメータは Oracle が順次スキャンの間に 1 つの I/O 操作に読み取るブロックの最大数を指定します。`/etc/system` のチューニングパラメータ `maxphys` はオペレーティングシステムレベルで最大の物理 I/O の転送サイズを確立します。最大の転送サイズを利用するには、Oracle の `init.ora` パラメータ `DB_FILE_MULTIBLOCK_READ_COUNT` を `maxphys/DB_BLOCK_SIZE` に設定する必要があります。

たとえば、`maxphys` が 1048576 に設定され、`DB_BLOCK_SIZE` が 4096 に設定されている場合、は 256 に設定する必要があります。DB_FILE_MULTIBLOCK_READ_COUNT

`DB_FILE_MULTIBLOCK_READ_COUNT` パラメータは Oracle が順次スキャンの間に 1 つの I/O 操作に読み取るブロックの最大数を指定します。
`DB_FILE_MULTIBLOCK_READ_COUNT` 初期化パラメータの値を大きくすると、通常、I/O スループットが向上します。AIX でこのパラメータは 1 から 512 まで指定できますが、一般的に、値を 16 より大きくしてもパフォーマンスはそれ以上は向上しません。

ファイルシステムがストライプボリュームで作成されるとき、このパラメータは DB_BLOCK_SIZE で除算したフルストライプサイズの倍数の値に設定します。フルストライプサイズを使うと、順次テーブルスキャンで読み取り操作がストライプディスクの完全な帯域幅を利用できるようになります。

DB_FILE_MULTIBLOCK_READ_COUNT を (read_pref_io*read_nstream)/DB_BLOCK_SIZE の倍数の値に設定しますが、この値は max_direct_iosz/DB_BLOCK_SIZE の値以下にする必要があります。

vxtunefs コマンドを使って read_pref_io、read_nstream、max_direct_iosz の値を表示します。例:

```
# vxtunefs /db01
```

vxtunefs コマンドにより、次のような出力が表示されます。

```
Filesystem i/o parameters for /db01

read_pref_io  = 65536
read_nstream  = 1
read_unit_io  = 65536
write_pref_io  = 65536
write_nstream  = 1
write_unit_io  = 65536
pref_strength  = 10
buf_breakup_size  = 262144
discovered_direct_iosz  = 262144
max_direct_iosz  = 1048576
default_indir_size = 8192

qio_cache_enable = 0

write_throttle = 0

max_diskq = 1048576

initial_extent_size = 8

max_seqio_extent_size = 2048

max_buf_data_size = 8192

hsm_write_prealloc = 0

read_ahead = 1

inode_aging_size = 0

inode_aging_count = 0
```

```
fcl_maxalloc = 130150400

fcl_keeptime = 0

fcl_winterval = 3600

oltp_load = 0
```

これらのパラメータとチューニングの説明について詳しくは、vxtunefs（1M）のマニュアルページを参照してください。

DB_FILE_MULTIBLOCK_READ_COUNT パラメータは Oracle が順次スキャンの間に 1 つの I/O 操作に読み取るブロックの最大数を指定します。/etc/system のチューニングパラメータ maxphys はオペレーティングシステムレベルで最大の物理 I/O の転送サイズを確立します。最大の転送サイズを利用するには、Oracle の init.ora パラメータ DB_FILE_MULTIBLOCK_READ_COUNT を maxphys/DB_BLOCK_SIZE に設定する必要があります。

たとえば、maxphys が 1048576 に設定され、DB_BLOCK_SIZE が 4096 に設定されている場合、は 256 に設定する必要があります。DB_FILE_MULTIBLOCK_READ_COUNT

スレーブパラメータの設定

Quick I/O および ODM はカーネルの非同期 I/O をサポートしており、複数の logwriter のスレーブやデータベースのライターのスレーブは必要としなくなりました。デフォルトではこのパラメータは 0 に設定されています。

Quick I/O を使用している場合は DBWR_IO_SLAVES の設定は必要ありません。DBWR のライターのプロセス数は DB_WRITER_PROCESSES 内で設定され、これにより非同期 I/O が実行されます。

ODM は非同期 I/O をサポートしており、複数の logwriter のスレーブやデータベースのライターのスレーブは必要としなくなりました。デフォルトではこのパラメータは 0 に設定されています。

メモリ割り当ての設定

システムで物理的に利用可能な容量を超えてメモリを使用するよう Oracle を設定しないでください。Oracle がシステムメモリリソースを求めて他のプロセスと競合せざるを得なくなる可能性があるため、メモリのサイズと割り当てを決める場合にはこれらの潜在的なプロセスのすべてを考慮する必要があります。理想的な設定では、Oracle 専用のシステムによりチューニングおよび監視に関する問題が単純化されて最適なパフォーマンスが確実に得られます。

AIX の仮想メモリマネージャのチューニングについて

Cached Quick I/O かバッファ付き I/O（つまり、Quick I/O または指定された拡張オプションのない単純な VxFS ファイル）のいずれかを使用している場合、データベースサーバーのスワップデバイスに対するすべてのページングのアクティビティを監視することをお勧めします。スワップデバイスのページングを監視するには、vmstat -I コマンドを使用します。スワップデバイスのページングに関する情報は、vmstat -I コマンド出力で pi および po と表示されたカラム（それぞれスワップデバイスからの読み取りと書き出し用）の下に表示されます。これらのカラムでゼロ以外の値が表示された場合、スワップデバイスのページングのアクティビティがあることが示唆されます。

次に例を示します。

```
# /usr/bin/vmstat -I

kthr      memory                 page                            faults     cpu
--------  --------------------- ----------------------------- ---------- -----------
r  b  p   avm     fre      fi   fo pi  po  fr    sr    in    sy     cs    us sy id wa

5  1  0   443602  1566524  661  20 0   0   7     28    4760  37401  7580  11 7  43 38

1  1  0   505780  1503791  18   6  0   0   0     0     1465  5176   848   1  1  97 1

1  1  0   592093  1373498  1464 1  0   0   0     0     4261  10703  7154  5  5  27 62

3  0  0   682693  1165463  3912 2  0   0   0     0     7984  19117  15672 16 13 1  70

4  0  0   775730  937562   4650 0  0   0   0     0     10082 24634  20048 22 15 0  63

6  0  0   864097  715214   4618 1  0   0   0     0     9762  26195  19666 23 16 1  61

5  0  0   951657  489668   4756 0  0   0   0     0     9926  27601  20116 24 15 1  60

4  1  0   1037864 266164   4733 5  0   0   0     0     9849  28748  20064 25 15 1  59

4  0  0   1122539 47155    4476 0  0   0   0     0     9473  29191  19490 26 16 1  57

5  4  0   1200050 247      4179 4  70  554 5300  27420 10793 31564  22500 30 18 1  52

6  10 0   1252543 98       2745 0  138 694 4625  12406 16190 30373  31312 35 14 2  49

7  14 0   1292402 220      2086 0  153 530 3559  17661 21343 32946  40525 43 12 1  44

7  18 0   1319988 183      1510 2  130 564 2587  14648 21011 28808  39800 38 9  3  49
```

スワップデバイスのページングを示す証拠がある場合、データベースパフォーマンスを向上させるために適切なAIXの仮想メモリマネージャ(VMM)のチューニングが必要になります。VMM のチューニングにより、ファイルシステムのキャッシュに割り当てられるメモリページの量を制限します。これにより、空きメモリ上で VMM の稼働率が低下したときにファイルシステムのキャッシュがアプリケーションからメモリのページを流用(その結果スワップデバイスの書き出しが発生)することがなくなります。

AIX VMM サブシステムをチューニングするコマンドは次のとおりです。

```
# /usr/samples/kernel/vmtune
```

vmtune によって行われた変更は、次にシステムを再ブートするときまで持続します。チューニングする VMM カーネルパラメータには、maxperm、maxclient、minperm があります。maxperm パラメータと maxclient パラメータは、ファイルシステムのキャッシュに使用できるメモリの最大量(メモリ合計のうちの割り合い)を指定します。ファイルシステムのキャッシュに使用するメモリの最大量は、AIX カーネルおよびすべてのアクティブなアプリケーションが残した未使用メモリの量を超えないように指定する必要があります。したがって、次のように計算できます。

100* (T-A) /T

ここで、T はシステム内のメモリページの合計数、A はすべてのアクティブなアプリケーションが使用するメモリページの最大数です。

minperm パラメータは、maxperm パラメータの値と等しいかそれ以下であり、かつ 5 以上に設定する必要があります。

AIX VMM のチューニングに関する詳細は、vmtune(1)のマニュアルページを参照してください。

次はチューニングが可能な VxFS の I/O パラメータです。

VMM バッファ件数

（-b <value> オプション）

仮想メモリマネージャバッファ（VMM）件数を設定します。VMM には、メモリの容量に基づくデフォルト値と現在の値との 2 つの値があります。vxtunefs -b を使用してこれら 2 つの値を表示できます。初期設定では、デフォルト値と現在の値は同じです。-b 値オプションは、デフォルト値から VMM バッファ件数の増分をゼロ % から 100 % で指定します。指定された値は、/etc/vx/vxfssystem ファイルに保存され、VxFS モジュールのロードやシステムの再ブート時にも永続します。

ほとんどの場合、良好なパフォーマンスを得るにはデフォルト値が適していますが、カーネルにあるカウンタで VMM バッファの待機時間に遅延が生じているかどうかを監視できます。VMM に関連するパフォーマンスの問題があるようであれば、バッファ件数は増加できます。システムで応答時間に改善がみられれば、VMM バッファにボトルネックがあったことが示唆されます。

kdb vmker コマンドで表示される次のフィールドは、ボトルネックの有無を判断するのに役立ちます。

```
THRPGIO buf wait (_waitcnt) value
```

このフィールドは、読み取りまたは書き出しが可能な VMM バッファがないことを示す可能性があります。このスレッドは、VMM バッファが使用可能となるのを待機して遮断されました。このカウントは、コールドロードからの合計待機数です。このフィールドは、kdb vmstat コマンドで表示される「読み取り」および「書き出し」のページとともに、VMM バッファ数が適切であるかどうかの判断に使用できます。次に比率を示します。

waitcnt / pageins+pageouts

この比率は、VMM バッファの待機数の指標となりますが、pageins + pageouts には他のファイルシステムのページ I/O やページングスペースが含まれるため厳密ではありません。この比率は、メモリ容量や VxFS 以外のファイルシステムへのページ I/O の量によって決まるため標準的な値というものはありません。0.1 を超える数値は VMM バッファ数のボトルネックを示唆する可能性があります。kdb vmker で表示されるその他の関連フィールドは次のとおりです。

- THRPGIO partial cnt (_partialcnt) value
  このフィールドは、利用可能な VMM バッファ数が I/O を必要とするページ数に満たないためにページ I/O が 2 つ以上の手順で行われたことを示します。
- THRPGIO full cnt (_fullcnt) value
  I/O を必要とする全ページに対してすべての VMM バッファが見つかったことを示します。

Oracle 用の Solaris のチューニングについて

Oracle に Storage Foundation を使って最適なパフォーマンスを実現するには、Solaris の特定のパラメータをチューニングする必要があります。これらのパラメータを変更するには、Solaris のカーネル設定（/etc/system ファイルで指定）を修正し、システムを再ブートする必要があります。

/etc/system ファイルのこれらのパラメータは、テキストエディタを使って追加または変更できます。次の例は、/etc/system の内容を示したものです。

```
* start Oracle *
set shmsys:shminfo_shmmax=0xffffffff
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=200
*
set semsys:seminfo_semmap=100
set semsys:seminfo_semmni=1000
set semsys:seminfo_semmns=4000
set semsys:seminfo_semmnu=800
set semsys:seminfo_semmsl=512
* end Oracle *
```

メモ: チューニング可能ないずれのパラメータの設定も、システムおよびデータベースのサイズ、データベース負荷、ユーザ数などの要因により異なります。場合によってパラメータを設定するよう提案することがありますが、いずれの場合にもご使用のシステムとバージョンに対応した Oracle のインストールガイドを参照し、Oracle 推奨の設定があればそれを使用してください。

---

maxuprc

このパラメータは 1 ユーザーによって同時に実行できるプロセスの最大数を設定します。データベースに対して同時に多数のユーザーがアクセスすることが予想されれば、このパラメータの値を増やす必要がある場合もあります。

maxuprc パラメータ値を増やすには

1. 次のようにして maxuprc の現在の設定を確認します。

   ```
   # echo "maxuprc/D" | adb -k
   ```

2. 次のようにして /etc/system ファイルの maxuprc の設定を修正または追加します。

   ```
   # set maxuprc=some_integer
   ```

shmmax

このパラメータは単一の共有メモリセグメントの最大サイズを(バイトで)設定します。推奨値についてはデータベースのマニュアルを参照してください。

shmmin

このパラメータは単一の共有メモリのセグメントの最小サイズをバイト単位で設定します。推奨値は Oracle のマニュアルを参照してください。

shmmni

このパラメータは共有メモリ識別子の数を設定します。推奨値についてはデータベースのマニュアルを参照してください。

shmseg

このパラメータはプロセスによって接続できる共有メモリセグメントの最大数を設定します。推奨値についてはデータベースのマニュアルを参照してください。

semmap

このパラメータはセマフォのマップのエントリ数を設定します。セマフォ作成のために与えられるこのメモリ領域は semmap から取得され、semmap の値に基づいてマップの一定のエントリ数で初期化されます。semmap の値は semmni より大きくすることはできません。推奨値についてはデータベースのマニュアルを参照してください。

semmni

このパラメータはシステムでのセマフォセット識別子の数を設定します。semmni パラメータにより一度に作成できるセマフォセットの数が決定され、大規模なデータベースにはこの値を高めに設定する必要がある場合があります。推奨値についてはデータベースのマニュアルを参照してください。

semmns

このパラメータはシステムでのセマフォの最大数を設定します。大規模なデータベースには semmns パラメータの値を高めに設定する必要がある場合があります。推奨値についてはデータベースのマニュアルを参照してください。

semmnu

このパラメータはシステム全体でのアンドゥ構造の最大数を設定します。このパラメータの値を semmni と等しく設定することにより、すべてのセマフォセットにアンドゥ構造が与えられます。semop(2) を使用して実行されるセマフォ操作はプロセスが終了した場合に元に戻すことができますが、アンドゥ構造は保証される必要があります。semmnu の推奨値についてはデータベースのマニュアルを参照してください。

semmsl

このパラメータは、1 つのセマフォのセットに含めることができるセマフォの最大数を設定します。semmsl パラメータは Oracle プロセスの最大数と同じである必要があります。推奨値は Oracle のマニュアルを参照してください。

30

SFDB ツールのトラブルシューティング

この章では以下の項目について説明しています。

- Storage Foundation for Databases（SFDB）ツールのトラブルシューティングについて
- vxdbd デーモンについて
- vxdbd のトラブルシューティング
- SFDB ツールのトラブルシューティングのためのリソース
- SmartTier for Oracle のトラブルシューティング
- SFDB（Storage Foundation for Databases）ツールの 5.0x から 7.4.2 へのアップグレード（2184482）
- 逆再同期のトラブルシューティング

Storage Foundation for Databases（SFDB）ツールのトラブルシューティングについて

Storage Foundation for Databases（SFDB）ツールは、複数の Storage Foundation 製品に配備されるため、これらの製品による問題の影響を受ける場合があります。問題発生時の最初の対応手順は、問題の原因を特定することです。Storage Foundation for Databases（SFDB）ツールで問題が発生するケースはまれです。多くの場合、原因はセットアップの問題にあるか、またはベース製品の問題にあります。

この章に記載された情報を使用して、問題の原因を診断してください。ベース製品のセットアップまたは設定の問題の可能性がある場合、他の Storage Foundation マニュアル

を参照して解決する必要がある場合があります。コンポーネント製品や Oracle が問題の原因である場合は、必要に応じて個別のマニュアルを参照してください。

Storage Foundation 製品の問題をトラブルシューティングする場合は、次のマニュアルを参照してください。

- Storage Foundation 管理者ガイド
- Storage Foundation for Cluster File System High Availability 管理者ガイド
- Storage Foundation for Oracle RAC 管理者ガイド

SFDB ツールのテクニカルサポート分析用スクリプトの実行

トラブルシューティングスクリプトによって、製品とその各モジュールの設定および状態に関する情報が収集されます。これらのスクリプトは、パッケージ情報、デバッグメッセージ、コンソールメッセージおよびディスクグループやボリュームの情報を識別します。これらのスクリプトの出力を Veritas テクニカルサポートに転送すると、問題の分析や解決に役立てることができます。

SFDB リポジトリとログ情報を入手するには

◆ 以下を実行します。

```
# /opt/VRTSspt/VRTSexplorer/VRTSexplorer
```

この出力をサポートに送信します。

Storage Foundation for Databases（SFDB）ツールのログファイル

次のログファイルを確認すると、有用な診断情報が得られます。

SFDB ツールコマンドのログファイルは /var/vx/vxdba/logs ディレクトリにあります。

vxdbd デーモンについて

SFDB のコマンドは、DBA ユーザーとして実行します。DBA ユーザーは、root ユーザーとして、スナップショットの作成やファイルシステムのマウントなどのいくつかの操作を実行する必要があります。vxdbd デーモンは、特権コマンドを実行するために、または別のホスト上にある SFDB リポジトリと通信するときに、SFDB コマンドが使用します。

vxdbd の起動と停止

vxdbd デーモンはシステム起動時に自動的に起動するように設定されています。/opt/VRTS/bin/sfae_config のスクリプトは、デーモンの停止と開始に使うことができますが、ステータスをクエリーするのにも使うことができます。デーモンを起動および停止できるのは root ユーザーのみです。

デーモンステータスをクエリーするには

◆ コマンドを実行します。

```
# /opt/VRTS/bin/sfae_config status
```

デーモンを起動するには

◆ コマンドを実行します。

```
# /opt/VRTS/bin/sfae_config enable
```

デーモンを停止するには

◆ コマンドを実行します。

```
# /opt/VRTS/bin/sfae_config disable
```

メモ: ほとんどの SFDB コマンドでは、vxdbd デーモンが実行されている必要があります。

---

vxdbd デーモンのリスニングポートの設定

vxdbd デーモンはデフォルトによって TCP ポート 3233 で応答準備をします。このポートが他の一部のアプリケーションで使用されている場合、vxdbd を設定して、代替ポートを応答準備できます。クラスタ環境では、すべてのノードの同じポートで応答準備するために、vxdbd を設定する必要があります。

vxdbd デーモンのリスニングポートを設定するには

1. vxdbd デーモンを停止します。

```
# /opt/VRTS/bin/sfae_config disable
```

2. /etc/vx/vxdbed/admin.properties 設定ファイルを編集することによって、VXDBD_PORT を必要な新しいポート番号に設定します。
3. vxdbd デーモンを起動します。

```
# /opt/VRTS/bin/sfae_config enable
```

vxdbd リソース使用量の制限

vxdbd デーモンはリソース使用量では軽いですが、システム管理者はさらにリソース使用量を制限するといいかもしれません。これは、/etc/vx/vxdbed/admin.properties の 2 つの設定ファイルを使用することによって制限できます。

- MAX_CONNECTIONS: このパラメータは vxdbd が許可すべき同時要求の最大数を制御します。
- MAX_REQUEST_SIZE: このパラメータは単一要求の最大サイズを制御します（バイト単位）。

これらのパラメータを低く設定しすぎると、SFDB コマンドが失敗する原因になる場合があります。次の内容は、これらのパラメータの推奨最小値です。

- MAX_CONNECTIONS: 5
- MAX_REQUEST_SIZE: 1048576（1 MB）

メモ: これらのパラメータに変更を加えたら、vxdbd を再起動して、変更内容を有効にします。

---

vxdbd の暗号化暗号文の設定

vxdbd デーモンとの通信は、暗号化されています。通信に使用される暗号化アルゴリズムは、`/var/vx/vxdba/auth/vxdbd/root/.VRTSat/profile/VRTSatlocal.conf` 設定ファイルを編集して設定できます。SSLCipherSuite 設定パラメータは、保護された接続を確立するときに許可される暗号文を指定します。このパラメータの受け入れ可能な値について詳しくは、OpenSSL ciphers(1) マニュアルページを参照してください。

たとえば、medium-strength および low-strength 暗号文を使用して通信を拒否するために `vxdbd` を設定するには、次のように SSLCipherSuite パラメータを設定します。

```
"SSLCipherSuite"="HIGH:!MEDIUM:!eNULL:!aNULL:!SSLv2:!LOW"
```

vxdbd のトラブルシューティング

`vxdbd` デーモンが動作していない場合、エラーメッセージを表示して SFDB コマンドが失敗する可能性があります。次の手順は、停止しているデーモンのインスタンスのトラブルシューティングに役立てることができます。

- `/opt/VRTS/bin/sfae_config status` を使って、デーモンが動作していることを確認します。
- 出力状態がデーモンが動作していないことを示す場合、`/opt/VRTS/bin/sfae_config enable` コマンドを使って、デーモンを起動します。
- デーモンが起動に失敗する場合、他のプロセスが同じポートで応答準備していないことを確認します。このようなプロセスがある場合は、プロセスを停止するか、`vxdbd` を設定して異なるポートで応答準備をします。

- デーモンは /var/vx/vxdba/logs/vxsfaed.log でファイルのログを生成します。詳細なログを有効にするには、/etc/vx/vxdbed/admin.properties 設定ファイルを編集して、LOG_LEVEL を DEBUG に設定します。

SFDB ツールのトラブルシューティングのためのリソース

SFDB（Storage Foundation for Database）のコマンドが失敗した場合は、次のリソースを使用してトラブルシューティングしてください。

p.372 の「SFDB ログ」を参照してください。

p.373 の「SFDB エラーメッセージ」を参照してください。

p.373 の「SFDB リポジトリとリポジトリファイル」を参照してください。

SFDB ログ

SFDB コマンドは、実際の問題の絞り込みに使えるログを生成します。

ログファイル:

- ログファイルは /var/vx/vxdba/logs の場所に生成されます。
- 2 つのタイプのログがあります。
  - ユーザーのログはユーザーのフォルダに生成されます。
  - vxdbd からのログおよび他のルート操作は、ログフォルダに生成されます。
- ユーザーのログファイルには命名規則があり、log_<service>_<app>_<service_id><app_id>.log という名前になります。system.log もまた、vxsfadm がサービスとアプリケーション ID を認識できるようになるまで存在します。
- vxdbd ログの名前は、vxsfaed.log です。system.log もまた実行されたすべてのルート操作に対して存在します。
- ログファイルは 1MB のしきい値に達した後にアーカイブされ、log_<service><application><application_identifier><service_identifier>.log としてバックアップされます。<randomnumber>
  各ログファイルには、前にアーカイブしたログに対するポインタがあります。

ログレベル:

- ログレベルは、SFAE_LOG_LEVEL 環境変数を使って設定できます。
- SFAE_LOG_LEVEL を上書きする次の環境変数をさらに設定できます。
  - APP_LOG_LEVEL: ログアプリケーション特有の操作

  - SER_LOG_LEVEL: ログ VxFS/VxVM スタック特有の操作
  - REP_LOG_LEVEL: ログリポジトリの操作
  - FSM_LOG_LEVEL: ログ vxsfadm エンジン特有の操作
- ログレベルは、次のレベルに設定できます。
  - FATAL: ログのみの致命的メッセージ
  - ERROR: ログエラーおよび上記のメッセージ
  - WARN: ログ警告および上記のメッセージ
  - INFO: ログ情報および上記のメッセージ
  - DEBUG: ログデバッグおよび上記のメッセージ
- デフォルトのログレベルは DEBUG です。

ログメッセージ:

- 実際のログメッセージが次の形式で表示されます。

  ```
  yyyy/mm/dd hh:mm:ss: <loglevel> : <module> : <message>
  ```

  次に例を示します。

SFDB エラーメッセージ

各エラーメッセージは、次のフィールドが表示される標準のテンプレートに基づいています。

- MESSAGE: 実際のエラーメッセージ
- REASON: エラーの理由
- ACTION: エラーの修正に行われるアクション

これらのフィールドは、問題の原因に関する正確な情報を提供します。

SFDB リポジトリとリポジトリファイル

注意: SFDB リポジトリに関連する操作のトラブルシューティングは、訓練を受けた Veritas 技術者の監督の下で行ってください。

リポジトリデータベースの名前は、`dbed-<application>-<application identifier>-repository.db` という形式になっています。

たとえば、`dbed-oracle-SFAEDB-repository.db` などです。

リポジトリデータベースをクエリーして、アプリケーションインスタンスのさまざまな情報を表示できます。これには、以下の表が含まれます。

- _fsm_state_: 各種のなサービスの進行状況を表示します。
- _operational_data_: さまざまなサービスのさまざまな設定値を表示します。
- files: サービスが使用するファイルを表示します。

SFDB ツールは、<repositorylocation>/files にファイルを作成します。

- これらのファイルは、すべてのサービスのさまざまな処理アクティビティに使用されます。
- リポジトリの files 表は、files ディレクトリの下にある、サービスが使用するさまざまなファイル名を指します。

SmartTier for Oracle のトラブルシューティング

SmartTier for Oracle コマンドが次の例のように失敗した場合は、tsdb_debug.log を確認します。

tsdb_debug.log は次の場所にあります。

```
/var/vx/vxdba/logs/tsdb_debug.log
```

たとえば、dbdst_addvol コマンドの実行後に次のメッセージが表示された場合は、tsdb_debug.log を確認してください。

```
$ /opt/VRTS/bin/dbdst_addvol -S BLM21 -M /dev/vx/dsk/nobody_data1/ ¥
data1vol -v new_vol1:fast_storage,new_vol2:slow_storage,new_vol3:slow_storage

SFORA dbdst_addvol ERROR V-81-6222 Could not add volume new_vol1 to vset
```

tsdb_debug.log ファイルには、次のような情報が含まれています。

```
# view /var/vx/vxdba/logs/tsdb_debug.log
1216606 Tue May 13 10:11:05 2008
 /opt/VRTS/bin/dbdst_addvol -S BLM21 -M
/dev/vx/dsk/nobody_data1/data1vol
-v new _vol1:fast_storage,new_vol2:slow_storage,
new_vol3:slow_storage
1216606 Tue May 13 10:11:09 2008
RACmaster = system1
1216606 Tue May 13 10:11:09 2008
system1:/opt/VRTSdbed/common/.dba/tsdb_setup.sh -g nobody_data1 -o
addvol
 -d¥ data1vol - v new_vol1 -m /oradata1 -t
```

```
vxfs.placement_class.FAST_STORAGE
1216606 Tue May 13 10:11:09 2008
command failed, ret=1

1216606 Tue May 13 10:11:09 2008
tsdb_setup.sh arguments -g nobody_data1 -o addvol -d data1vol -v
 new_vol1 -m /or
adata1 -t vxfs.placement_class.FAST_STORAGE
05/13/08@17:08:11
size of volume  new_vol1 is 204800
VxVM vxvset ERROR V-5-1-10035 Volume set data1vol contains volume(s)

in snapshot chain.
This can cause inconsistencies in the snapshot hierarchy. Specify
"-f"
option to force the operation.
^^^^ NOTE: here is the reason for the failure, barried in this log
file:
^^^^ /var/vx/vxdba/logs/tsdb_debug.log
Can not add  to data1vol, ERR 1
ERROR:1
```

SFDB（Storage Foundation for Databases）ツールの 5.0x から 7.4.2 へのアップグレード（2184482）

SFHA または SF for Oracle RAC のバージョン 5.0 または 5.0MP3 を SFHA または SF for Oracle RAC のバージョン 7.4.2 にアップグレードすると、sfua_rept_migrate コマンドによってエラーメッセージが表示されます。

SFHA または SF for Oracle RAC のバージョン 5.0 を SFHA または SF for Oracle RAC のバージョン 7.4.2 にアップグレードすると、sfua_rept_migrate コマンドによってエラーメッセージが表示されます。

SF Enterprise 製品 バージョン 5.0 または 5.0MP3 から SF Enterprise 製品 7.4.2 にアップグレードする際に、S*vxdbms3 起動スクリプトは NO_S*vxdbms3 に名前変更されます。sfua_rept_upgrade では S*vxdbms3 起動スクリプトが必要です。このため、sfua_rept_upgrade が実行されると、S*vxdbms3 起動スクリプトが見つからず、上記のエラーメッセージが表示されます。

```
/sbin/rc3.d/S*vxdbms3 not found
SFORA sfua_rept_migrate ERROR V-81-3558 File:  is missing.
SFORA sfua_rept_migrate ERROR V-81-9160 Failed to mount repository.
```

回避策:`sfua_rept_migrate` を実行する前に、起動スクリプト名 NO_S*vxdbms3 を S*vxdbms3 に変更します。

逆再同期のトラブルシューティング

バックアップデータベースインスタンスのプレビューセッションを準備する前に、逆再同期の開始操作は常にリカバリポイントを作成します。このリカバリポイントが失われると、逆再同期の開始状態から移行できません。

次の手順は、インスタンスをトラブルシューティングし、逆再同期の開始状態に移行する場合に役立ちます。

1. ［SQL］ > ［Shutdown immediate］を使って Oracle データベースインスタンスを停止します。
2. すべてのスナップショットボリュームのマウントを解除します。スナップショットのボリューム名には、スナップショット設定ファイルに指定されているスナップショットの接頭辞が先頭に付けられます。接頭辞のデフォルト値は "SNAP_" です。

   次に例を示します。

   ```
   # umount /data/prodDB
   ```

3. スナップショットのディスクグループをデポートします。スナップショットのディスクグループ名には、スナップショット設定ファイルに指定されているスナップショットの接頭辞が先頭に付けられます。

   次に例を示します。

   ```
   # vxdg deport SNAP_proddg
   ```

4 プライマリデータボリュームをマウントします。

次に例を示します。

```
#mount -t vxfs /dev/vx/dsk/proddg/dbvol1 /data/prodDB
```

5 バックアップリポジトリをリストアして前の状態に切り替えます。

RRBegin 操作は、SFDB リポジトリを自動的にバックアップします。

```
[oracle@testbox ~]$ sfua_rept_util -S $ORACLE_SID -o list
Listing Repository Backups for SID='RR'
TIME                    SIZE            NAME
2013-07-30-13-43-55     18160
rept_RR_2013-07-30-13-43-55.tgz
2013-07-30-16-16-25     18159
rept_RR_2013-07-30-16-16-25.tgz
2013-07-30-16-16-49     18161
rept_RR_2013-07-30-16-16-49.tgz
2013-07-30-16-17-56     18163
rept_RR_2013-07-30-16-17-56.tgz
2013-07-30-16-18-01     18163
rept_RR_2013-07-30-16-18-01.tgz
2013-07-30-16-18-07     18161
rept_RR_2013-07-30-16-18-07.tgz

[oracle@testbox ~]$ sfua_rept_util -S $ORACLE_SID -o restore -f
rept_RR_2013-07-30-16-18-07.tgz
Restoring repository for SID='RR'
Target directory='/etc/vx/vxdba/logs/oracle/backup'
Restore Repository for SID='RR' Completed
```

31

Oracle データベースの手動によるリカバリ

この章では以下の項目について説明しています。

- クローンデータベースの手動による回復
- 復元操作の後の手動による回復

クローンデータベースの手動による回復

Storage Foundation for Databases（SFDB）ツールでは、ユーザーは PITC に基づいたアプリケーションをクローンできます。

アプリケーションのクローンに必要なアプリケーションレベルの回復は、オンラインおよびインスタントモードのスナップショットの場合、通常、クローン操作が実行されるたびに自動的に行われます。ただし、クローンデータベースインスタンスに対しては手動で Oracle 回復を実行した方がよいかもしれません。 手動による回復は、スナップショットが作成された時間の後のポイントに、クローンデータベースの PIT 回復を実行するなどの理由から必要になる可能性があります。

手動による回復を実行するには、クローン操作の代わりに preclone または mountdb 操作を使用します。preclone 操作はクローンデータベースをマウントしており、ユーザーはデータベース回復を手動で実行して、データベースを開くことができます。

メモ: オンラインのスナップショットモードを使用している場合は、手動による回復オプションだけが使用可能で、推奨されます。オフラインのスナップショットには回復は必要なく、データベースは RESETLOGS を使用して直接開くことができます。

次の一連の SQL 文を使用して、手動による回復を実行します。

```
1) ALTER DATABASE RECOVER UNTIL CANCEL USING BACKUP CONTROLFILE;
 OR

 ALTER DATABASE RECOVER UNTIL TIME <time> USING BACKUP CONTROLFILE;
 OR

 ALTER DATABASE RECOVER UNTIL CHANGE <SCN> USING BACKUP CONTROLFILE;

2) ALTER DATABASE RECOVER LOGFILE '<archived redo log file>';
{ Repeat this step until the desired recovery point/target is
reached.}
{The recovery target must be after the time the snapshot was taken.}

3) ALTER DATABASE RECOVER CANCEL;
{ Use if cancel-based recovery was initiated in step 1.}
```

手動による回復が終了したら、RESETLOGS オプションを使用してデータベースを開くことができます。その後、読み取り専用またはオフラインの表スペースをクローンデータベースに再度追加できるように、読み取り専用またはオフラインの表スペースに属するデータファイルを追加できます。一時ファイルもクローンデータベースに追加して、一時表スペースのストレージを提供できます。

復元操作の後の手動による回復

Storage Foundation for Databases（SFDB）ツールの Storage Checkpoint と FileSnap サービスは、PITC からアプリケーションファイルを復元する復元操作を提供します。Oracle の場合、復元操作は PITC からデータベースファイルを復元します。

復元操作は、プライマリデータベースデータファイルにデータファイルのみを復元し、上書きします。その後、ユーザーは手動でデータベース PIT 回復を実行して、必要な回復ターゲットまたは時間にデータベースを回復できます。復元操作は、またスナップショットから制御ファイルのコピーを /var/tmp ディレクトリに配置して、ユーザーがスナップショット時間に PIT 回復を実行できるようにします。

メモ: オンラインまたはオフラインモードのスナップショットでのみ、復元操作を実行できます。

復元操作を実行した後、次の SQL 文を使って、スナップショットが行われた時間からデータベースを任意の PIT に回復します。

```
{ To perform complete recovery:}
1) SET AUTORECOVERY ON;
```

```
2) RECOVER DATABASE;

{ To perform incomplete recovery:}

1) SET AUTORECOVERY ON;

2) RECOVER DATABASE UNTIL CANCEL;

OR

RECOVER DATABASE UNTIL TIME <time>;

OR

RECOVER DATABASE UNTIL CHANGE <SCN>;

{ Note: The 'USING BACKUP CONTROLFILE' clause must be used
if recovery is performed using a backup control file, or if using
the control file from the snapshot.}

3) ALTER DATABASE OPEN RESETLOGS;
```

32

6.0 より前のリリースの Storage Foundation for Databases のコマンドリファレンス

この章では以下の項目について説明しています。

- SFDB コマンドの後方互換について
- サポート対象外となった Storage Foundation for Databases（SFDB）ツールの機能
- Database FlashSnap のストレージの準備
- データベーススナップショットの作成について
- FlashSnap コマンド
- Oracle リカバリのガイドライン
- Database Storage Checkpoint のコマンド

SFDB コマンドの後方互換について

次の項には以前のリリースからの Storage Foundation for Databases（SFDB）ツールのコマンドについての情報が含まれています。`vxsfadm` CLI の使用をお勧めします。以前のリリースのコマンドは後方互換性がサポートされますが、今後のリリースでは非推奨になります。

サポート対象外となった Storage Foundation for Databases（SFDB）ツールの機能

次の SFDB（Storage Foundation for Database）ツールの機能は、このリリースではサポートされていません。

- Storage Checkpoint ポリシーと Storage Checkpoint クォータ
- クローンとロールバックの対話モード

Database FlashSnap のストレージの準備

次の項には、Database FlashSnap のストレージの準備についての情報が含まれます。

ボリュームのスナップショットミラーを作成する手順例

次の例は、ディスクグループ PRODdg に属するボリューム data_vol のスナップショットミラーを作成する手順を示しています。

メモ: 次の手順でコマンドを実行するには、スーパーユーザー（root）としてログインする必要があります。

ボリューム data_vol のスナップショットミラーを作成するには

1. ボリューム data_vol をミラー化できるように準備します。

```
# vxsnap -g PRODdg prepare data_vol alloc=PRODdg01,PRODdg02
```

2. FastResync が有効であることを確認するには、次のように入力します。

```
# vxprint -g PRODdg -F%fastresync data_vol

on
```

3. DCO と DCO ログがボリュームに関連付けられていることを確認するには、次を実行します。

```
# vxprint -g PRODdg -F%hasdcolog data_vol

on
```

4. data_vol のスナップショットミラーを作成するには、次のように入力します。

```
# vxsnap -g PRODdg addmir data_vol alloc=PRODdg02
```

5 データプレックスを一覧で表示するには、次のように入力します。

```
# vxprint -g PRODdg -F%name -e"pl_v_name in ¥"data_vol¥""

data_vol-01

data_vol-02
```

6 SNAPDONE 状態になっているプレックスを選択します。`vxprint -g diskgroup` コマンドを使って、SNAPDONE 状態になっているプレックスを識別します。

7 上記の手順でプレックス名を識別し、そのデータプレックスに dbed_flashsnap タグを設定するには、次のように入力します。

```
# vxedit -g PRODdg set putil2=dbed_flashsnap data_vol-02
```

8 `dbed_flashsnap` タグが、必要なデータプレックス data_vol-02 に設定されていることを確認するには、次のように入力します。

```
# vxprint -g PRODdg -F%name -e"pl_v_name in ¥"data_vol¥" ¥
&& p2 in ¥"dbed_flashsnap¥""
data_vol-02
```

9 スナップショットボリュームの作成に成功したことを確認するには、vxprint -g dg コマンドを次のように実行します。

```
# vxprint -g PRODdg

# vxprint -g flashdgso

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg flashdgso    flashdgso    -        -        -        -        -       -

dm hdisk5       hdisk5       -        10411776 -        -        -       -
dm hdisk6       hdisk6       -        10411776 -        -        -       -
dm hdisk8       hdisk8       -        10411776 -        -        -       -
dm hdisk10      hdisk10      -        10411776 -        -        -       -
dm hdisk14      hdisk14      -        10411776 -        -        -       -
dm hdisk17      hdisk17      -        10411776 -        -        -       -

v  archvol      fsgen        ENABLED  10237952 -        ACTIVE   -       -
pl archvol-01   archvol      ENABLED  10237952 -        ACTIVE   -       -
sd hdisk6-01    archvol-01   ENABLED  10237952 0        -        -       -
pl archvol-02   archvol      ENABLED  10237952 -        SNAPDONE -       -
sd hdisk8-01    archvol-02   ENABLED  10237952 0        -        -       -
dc archvol_dco  archvol      -        -        -        -        -       -
v  archvol_dcl  gen          ENABLED  1120     -        ACTIVE   -       -
pl archvol_dcl-01 archvol_dcl ENABLED 1120     -        ACTIVE   -       -
sd hdisk14-01   archvol_dcl-01 ENABLED 1120    0        -        -       -
pl archvol_dcl-02 archvol_dcl DISABLED 1120    -        DCOSNP   -       -
sd hdisk8-02    archvol_dcl-02 ENABLED 1120    0        -        -       -

v  datavol      fsgen        ENABLED  10237952 -        ACTIVE   -       -
pl datavol-01   datavol      ENABLED  10237952 -        ACTIVE   -       -
sd hdisk5-01    datavol-01   ENABLED  10237952 0        -        -       -
pl datavol-02   datavol      ENABLED  10237952 -        SNAPDONE -       -
sd hdisk10-01   datavol-02   ENABLED  10237952 0        -        -       -
dc datavol_dco  datavol      -        -        -        -        -       -
v  datavol_dcl  gen          ENABLED  1120     -        ACTIVE   -       -
pl datavol_dcl-01 datavol_dcl ENABLED 1120     -        ACTIVE   -       -
sd hdisk17-01   datavol_dcl-01 ENABLED 1120    0        -        -       -
pl datavol_dcl-02 datavol_dcl DISABLED 1120    -        DCOSNP   -       -
sd hdisk10-02   datavol_dcl-02 ENABLED 1120    0        -        -       -

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0 PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     Disk_1       -        71117760 -        -        -       -
```

```
dm PRODdg02     Disk_2       -        71117760 -        -        -       -
dm PRODdg03     Disk_3       -        71117760 -        -        -       -

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -

      v  data_vol     fsgen
      ENABLED        4194304             -        ACTIVE   -       -
      pl data_vol-01  data_vol
      ENABLED        4194304             -        ACTIVE   -       -
      sd PRODdg03-01  data_vol-01
      ENABLED        4194304             0        -        -       -
      pl data_vol-02  data_vol
      ENABLED        4194304             -        SNAPDONE -       -
      sd PRODdg02-01  data_vol-02
      ENABLED        4194304             0        -        -       -
      dc data_vol_dco data_vol
      -              -                   -        -        -       -
      v  data_vol_dcl gen
      ENABLED        560                 -        ACTIVE   -       -
      pl data_vol_dcl-01 data_vol_dcl ENABLED
      560         -           ACTIVE   -       -
      sd PRODdg01-01  data_vol_dcl-01 ENABLED
      560         0           -        -       -
      pl data_vol_dcl-02 data_vol_dcl DISABLED
      560         -           DCOSNP   -       -
      sd PRODdg02-02  data_vol_dcl-02 ENABLED
      560         0           -        -       -
```

指定したプレックスの状態が SNAPDONE であることを識別します。この例では、プレックスが data_vol-02 になります。

スナップショットミラーが使える状態になりました。

Veritas Volume Manager 7.4.2 を使うための既存ボリュームのアップグレード

この項の手順では、Database FlashSnap および領域最適化スナップショットの利点を活用できるようにするため、VxVM 5.0 よりも古いバージョンで作成したボリュームをアップグレードする方法を説明します。

この手順では、次の必要条件と通告に注意してください。

- DCO ボリュームのプレックスでは、ディスク上の永続ストレージ領域が使える状態にある必要があります。DCO プレックスで使う領域を用意するために、ディスクグループにさらにディスクを追加するか、既存のボリュームを再設定しディスクグループ内の領域を解放する必要がある場合があります。ディスク領域を追加する別の方法は、ディスクグループの移動機能を使って別のディスクグループからスペアディスクを持ってくることです。
- `vxassist` コマンドで作成した既存のスナップショットボリュームは対応していません。`vxassist` と `vxsnap` で作成したスナップショットボリュームの組み合わせもサポートされていません。
- 次の手順でコマンドを実行するには、スーパーユーザー（root）としてログインする必要があります。

以前のバージョンの VxVM で作成された既存のボリュームをアップグレードするには

1. ここで説明する手順の残り部分を実施する前に、ボリュームを含むディスクグループをバージョン 120 以上にアップグレードします。ディスクグループのバージョンを確認するには、次のコマンドを使います。

   ```
   # vxdg list diskgroup
   ```

   ディスクグループを最新バージョンにアップグレードするには、次のコマンドを使います。

   ```
   # vxdg upgrade diskgroup
   ```

2. アップグレードしたボリュームに、以前のバージョンの VxVM の DRL プレックスまたはサブディスクが含まれている場合は、次のコマンドを使ってそれを削除します。

   ```
   # vxassist [-g diskgroup] remove log volume [nlog=n]
   ```

   削除するログ数n を指定するには、属性 nlog=*n*（オプション）を使用します。デフォルトでは、`vxassist` コマンドで削除されるログは 1 つです。

3. 1 つ以上のスナップショットボリュームが関連付けられているボリュームの場合は、次のコマンドを使って各スナップショットの再接続および再同期を行います。

   ```
   # vxsnap [-g diskgroup] snapback snapvol
   ```

   スナップショットを取得する前に、永続 FastResync が有効であった場合は、スナップショットプレックスのデータをもとのボリュームから早急に再同期化させます。永続 FastResync が有効でなかった場合は、完全再同期処理が実行されます。

4. ボリュームの永続 FastResync を無効にするには、次のコマンドを実行します。

   ```
   # vxvol [-g diskgroup] set fastresync=off volume
   ```

5 次のコマンドを使って、VxVM（以前のバージョン）と DCO オブジェクト、ボリュームと DCO ボリューム、ボリュームとスナップオブジェクトの 3 つの関連付けを解除します。

```
# vxassist [-g diskgroup] remove log volume logtype=dco
```

6 ボリュームで次のコマンドを使って、ボリュームをアップグレードします。

```
# vxsnap [-g diskgroup] prepare volume ¥
alloc="disk_name1,disk_name2"
```

スナップショット DCO プレックスの領域と、その他の移動していないデータや DCO プレックスの領域が重ならないようにディスク名を 2 つ用意します。

vxsnap prepare コマンドは、ボリュームとボリュームから生成されたすべてのスナップショットに対する永続 FastResync を自動的に有効にします。またこのコマンドでは、DCO や DCO ログボリュームと、スナップショットであるボリュームを関連付けます。

7 既存の DCO プレックスを表示し、既存のデータプレックスに十分な DCO プレックスがあることを確認するには、次を入力します。

```
# vxprint -g diskgroup
```

既存の各データプレックスにそれぞれ 1 つの DCO プレックスがなくてはなりません。

8 既存のデータプレックスに十分な数の DCO プレックスがない場合は、次のコマンドを使って DCO プレックスを作成してその数を増やします。

```
# vxsnap [-g diskgroup] addmir dco_volume_name ¥
[alloc=disk_name]
```

dco_volume_name は、作成する DCO ボリュームの名前です。

9 プレックスが SNAPDONE 状態の場合は、ACTIVE 状態に変換します。

```
# vxplex [-g diskgroup] convert state=ACTIVE data_plex
```

10 データプレックスを SNAPDONE の状態に変更し、DCO プレックスと、スナップショットの操作に使うデータプレックスを関連付けるには、次のように入力します。

```
# vxplex [-g diskgroup] -o dcoplex=dco_plex_name convert ¥
state=SNAPDONE data_plex
```

dco_plex_name は、作成する DCO プレックスの名前です。

Veritas Volume Manager 7.4.2 を使うための既存ボリュームのアップグレードの手順例

メモ: 次の手順でコマンドを実行するには、スーパーユーザー（root）としてログインする必要があります。

この例では、VxVM 7.4.2 の機能が使えるようにボリューム `data_vol` をアップグレードします。

以前のバージョンの VxVM で作成された既存のボリュームをアップグレードするには

1. ディスクグループ PRODdg をアップグレードします。

```
# vxdg upgrade PRODdg
```

2. アップグレードするボリュームから、以前のバージョンの VxVM に所属する DRL プレックスまたはサブディスクを削除します。

```
# vxassist -g PRODdg remove log data_vol logtype=drl
```

3. スナップショットボリュームを、アップグレードするプライマリボリュームに再度関連付けます。

```
# vxsnap -g PRODdg snapback SNAP-data_vol
```

4. アップグレードするボリュームの FastResync を無効にします。

```
# vxvol -g PRODdg set fastresync=off data_vol
```

5. 古い DCO オブジェクトと DCO ボリュームをすべて関連付けを解除し、削除します。

```
# vxassist -g PRODdg remove log data_vol logtype=dco
```

6. 新しい DCO オブジェクトと DCO ボリュームを関連付け、ボリュームをアップグレードします。

```
# vxsnap -g PRODdg prepare data_vol alloc="PRODdg01,PRODdg02"
```

7. 既存の DCO プレックスとプレックスの状態を表示します。

シナリオ 1

このシナリオでは、データプレックスに十分な DCO プレックスが存在するものとします。また、DCO プレックスに関連付けられているデータプレックスは存在しません。

```
# vxprint -g PRODdg


TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -

v  data_vol     fsgen
ENABLED        4194304          -        ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED        4194304          -        ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED        4194304          0        -        -       -
pl data_vol-04  data_vol
ENABLED        4194304          -        SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED        4194304          0        -        -       -
dc data_vol_dco data_vol
-              -                -        -        -       -
v  data_vol_dcl gen
ENABLED       560               -        ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED       560               -        ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED       560               0        -        -       -
pl data_vol_dcl-02 data_vol_dcl
ENABLED       560               -        ACTIVE   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED       560               0        -        -       -
```

- データプレックスの状態を SNAPDONE から ACTIVE に変更します。

```
# vxplex -g PRODdg convert state=ACTIVE data_vol-04
```

- データプレックスと新しい DCO プレックスとの関連付けを行い、そのデータプレックスの状態を SNAPDONE に戻します。

```
# vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 ¥
convert state=SNAPDONE data_vol-04

# vxprint -g PRODdg


TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -

pl data_vol-03  -
DISABLED       4194304          -         -        -       -
sd PRODdg02-01  data_vol-03
ENABLED        4194304          0         -        -       -

v  data_vol     fsgen
ENABLED        4194304          -         ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED        4194304          -         ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED        4194304          0         -        -       -
pl data_vol-04  data_vol
ENABLED        4194304          -         SNAPDONE -       -
sd PRODdg02-03  data_vol-04
ENABLED        4194304          0         -        -       -
dc data_vol_dco data_vol
-              -                -         -        -       -
v  data_vol_dcl gen
ENABLED        560              -         ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED        560              -         ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED        560              0         -        -       -
pl data_vol_dcl-02 data_vol_dcl
DISABLED       560              -         DCOSNP   -       -
sd PRODdg02-02  data_vol_dcl-02
ENABLED        560              0         -        -       -
```

シナリオ 2

このシナリオでは、データプレックスよりも DCO プレックスの数が少ないものとします。

```
# vxprint -g PRODdg

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -

pl data_vol-03  -
DISABLED       4194304       -        -        -       -
sd PRODdg02-01  data_vol-03
ENABLED        4194304       0        -        -       -
v  data_vol     fsgen
ENABLED        4194304       -        ACTIVE   -       -
pl data_vol-01  data_vol
ENABLED        4194304       -        ACTIVE   -       -
sd PRODdg01-01  data_vol-01
ENABLED        4194304       0        -        -       -
pl data_vol-04  data_vol
ENABLED        4194304       -        ACTIVE   -       -
sd PRODdg02-03  data_vol-04
ENABLED        4194304       0        -        -       -
dc data_vol_dco data_vol
-              -             -        -        -       -
v  data_vol_dcl gen
ENABLED        560           -        ACTIVE   -       -
pl data_vol_dcl-01 data_vol_dcl
ENABLED        560           -        ACTIVE   -       -
sd PRODdg01-02  data_vol_dcl-01
ENABLED        560           0        -        -       -
```

- vxassist mirror コマンドを使って、DCO プレックスを DCO ボリュームに追加します。

```
# vxsnap -g PRODdg addmir data_vol_dcl alloc=PRODdg02
```

- データプレックスと新しい DCO プレックスとの関連付けを行い、そのデータプレックスの状態を SNAPDONE に変更します。

```
# vxplex -g PRODdg -o dcoplex=data_vol_dcl-02 ¥
  convert state=SNAPDONE data_vol-04
```

```
TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg PRODdg       PRODdg       -        -        -        -        -       -
dm PRODdg01     c1t2d0s2     -        35358848 -        -        -       -
dm PRODdg02     c1t3d0s2     -        17674896 -        -        -       -
dm PRODdg03     c1t1d0s2     -        17674896 -        -        -       -

                         pl data_vol-03  -
                         DISABLED       4194304       -        -        -       -
                         v  data_vol     fsgen
                         ENABLED        4194304       -        ACTIVE   -       -
                         pl data_vol-01  data_vol
                         ENABLED        4194304       -        ACTIVE   -       -
                         sd PRODdg01-01  data_vol-01
                         ENABLED        4194304       0        -        -       -
                         pl data_vol-04  data_vol
                         ENABLED        4194304       -        SNAPDONE -       -
                         sd PRODdg02-03  data_vol-04
                         ENABLED        4194304       0        -        -       -
                         dc data_vol_dco data_vol
                         -              -             -        -        -       -
                         v  data_vol_dcl gen
                         ENABLED        560           -        ACTIVE   -       -
                         pl data_vol_dcl-01 data_vol_dcl
                         ENABLED        560           -        ACTIVE   -       -
                         sd PRODdg01-02  data_vol_dcl-01
                         ENABLED        560           0        -        -       -
                         pl data_vol_dcl-02 data_vol_dcl
                         DISABLED       560           -        DCOSNP   -       -
                         sd PRODdg02-02  data_vol_dcl-02
                         ENABLED        560           0        -        -       -
```

データベースの構造変更後の更新

制御ファイル、データファイル、arch の位置といったデータベースオブジェクトの追加や削除など、データベースに対して何らかの構造的な変更を行う場合は、ノード上で dbed_update を実行して、SFDB リポジトリを更新する必要があります。データベースの変更前に作成されたフラッシュスナップがある場合は、次の手順を実行する必要があります。

データベースの構造変更後にフラッシュスナップを更新するには

1. dbed_vmsnap ユーティリティを使ってスナップショットを再同期します。
2. dbed_update を実行します。
3. 再度スナッププランを検証し、新しいスナップショットを作成します。

データベーススナップショットの作成について

スナップショットは、データベースのバックアップ元にしたり、意思決定支援システム用のクローンデータベースの作成元にできます。Database FlashSnap コマンドを使って、同じホスト(ノード)上または別のホスト上においてデータベース全体にわたるスナップショットを作成できます。

オンラインデータベーススナップショット

「表 32-1」では、作成できる 3 タイプのスナップショットについて説明します。

表 32-1　データベーススナップショットのタイプ

データベーススナップショットのタイプ	説明
online	スナッププランに指定された SNAPSHOT_MODE が online に設定されている場合、dbed_vmsnap コマンドにより、まずスナップショットである表領域がバックアップモードになります。スナップショットが作成された後、表領域はバックアップモードではなくなり、追加の REDO ログがアーカイブ化されるようにログファイルが切り替えられ、スナップショットのアーカイブログが作成されます。 オンラインスナップショットとオフラインスナップショットは、データベースの有効なバックアップコピーを提供します。
offline	SNAPSHOT_MODE を offline に設定した場合は、スナップショットを作成する前にデータベースが停止している必要があります。オンライン REDO ログとコントロールファイルが必要で、これらはデータベースの完全なリカバリを確実にするために使われます。
instant	SNAPSHOT_MODE を instant に設定した場合は、表領域はバックアップモードになったり、バックアップモードから解除されたりすることはありません。オンライン REDO ログとコントロールファイルが必要で、これらはデータベースの完全なリカバリを確実にするために使われます。 インスタントスナップショットは、PIT リカバリの有効なバックアップコピーにはなりません。

メモ: Storage Foundation for Oracle RAC では、オンラインスナップショットモードのみサポートされます。

Database FlashSnap は、オンラインのデータベーススナップショットのタイプをサポートします。

スナッププランに指定された SNAPSHOT_MODE が online に設定されている場合、`dbed_vmsnap` コマンドにより、まずスナップショットである表領域がバックアップモードになります。スナップショットが作成された後、表領域はバックアップモードではなくなり、追加の REDO ログがアーカイブ化されるようにログファイルが切り替えられ、スナップショットのアーカイブログが作成されます。

オンラインのスナップショットは、データベースの有効なバックアップコピーを提供します。

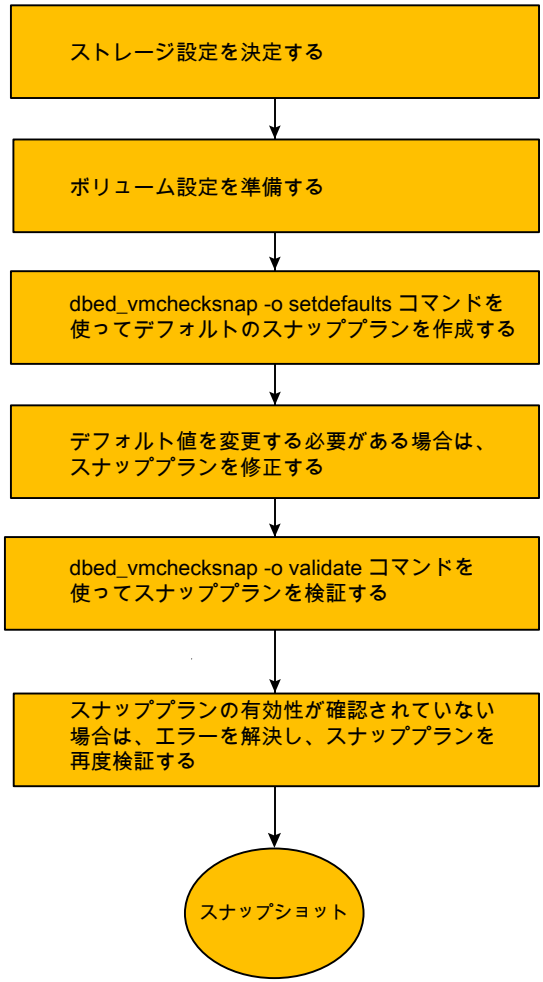
スナップショット作成前の作業

データベース上のすべてのボリュームのスナップショットをスナッププランを使って作成する詳しい方法について確認します。

オプションにより、VxVM コマンド（`vxsnap`）を使ってボリュームのスナップショットを作成できます。ただし、Database FlashSnap のコマンドとは異なる `vxsnap` コマンドを使っても、ディスクグループの内容を再認識させる機能は自動化されません。

図 32-1 は、Database FlashSnap を使ってスナップショットを作成するまでの一連の手順を示しています。

図 32-1　データベースのスナップショットを作成する際の前提条件



スナップショットの作成

スナップショットを取得する前にデータベースで使うボリュームが正しく設定されていることを確認します。このデータベース設定には、スーパーユーザー（root）権限が必要です。

メモ: Database FlashSnap コマンドの実行は、Oracle データベース管理者が行う必要があります。

データベースの構造を変更するとき(たとえば、データファイルの追加、削除、PFILE から SPFILE への変換、SPFILE から PFILE への変換)、dbed_update を実行する必要があります。次に例を示します。

```
$ /opt/VRTS/bin/dbed_update -S $ORACLE_SID -H $ORACLE_HOME
```

データベースのスナップショットイメージを作成するには

1. ボリュームまたはボリュームセットのスナップショットミラーを作成します。

   p.193 の「ボリュームまたはボリュームセットのスナップショットミラーを作成するには」を参照してください。

2. dbed_vmchecksnap コマンドを使って、スナッププランのテンプレートを作成し、データベースのボリュームスナップショットを作成するうえでそのテンプレートが有効となるようにボリュームの設定をチェックします。

   スナッププランには、細分化されたデータベースや、スナップショットの作成や再同期処理に必要なボリューム設定情報が含まれています。スナッププランのテンプレートは、テキストエディタを使って編集できます。

   dbed_vmchecksnap コマンドは、次の操作にも使えます。

特定の $ORACLE_SID に関連付けられたすべてのスナッププランを一覧表示します。	dbed_vmchecksnap -o list
スナッププランを SFDB リポジトリから削除します。	dbed_vmchecksnap -o remove -f SNAPPLAN
SFDB レポジトリからローカルディレクトリにスナッププランをコピーします。	dbed_vmchecksnap -o copy -f SNAPPLAN

3. dbed_vmsnap コマンドを使って、データベースのスナップショットボリュームを作成します。

   p.190 の「データベースによって使われるボリュームまたはボリュームセットのスナップショットミラーの作成」を参照してください。

4. セカンダリホストでは、dbed_vmclonedb コマンドを使って、プライマリホストからデポートされたディスクグループを使うクローンデータベースを作成します。詳しくは次を参照してください。

   p.205 の「セカンダリホストでのデータベースのクローン作成」を参照してください。

   スナッププランで指定したプライマリホストとセカンダリホストが異なる場合、dbed_vmclonedb コマンドを使って次の操作を実行します。

   - プライマリホストからデポートされたディスクグループをインポートする
   - スナップショットボリュームのマウントをリカバリする

- ファイルシステムをマウントする
- データベースをリカバリする
- プライマリホストとは異なる Oracle SID 名でデータベースをオンラインにする

`-o recoverdb` オプションを使って、`dbed_vmclonedb` にデータベースのリカバリを自動で実行させるか、`-o mountdb` オプションを使って、独自の PIT リカバリを実行してデータベースを手動で起動させることができます。PIT リカバリでは、スナップショットのモードが online である必要があります。

また、プライマリホストにおいてクローンも作成できます。スナッププランの設定では、クローンをプライマリホストまたはセカンダリホストのいずれかで作成するかを指定します。

5 これで、クローンデータベースを使って、データベースのバックアップや他のオフホスト処理作業が行えるようになりました。

6 単一インスタンス Oracle では、スナップショットボリュームは、スナップショットのデータから元のボリュームを逆再同期化させるのに使うこともできるほか、今後の使用に備えて、元のボリュームにスナップショットボリュームを再結合することにより（つまり、スナップショットボリュームを再同期化することにより）、破棄することもできます。

7 クローンデータベースは、今後の使用に備えて、元のボリュームにスナップショットボリュームを再結合することにより（つまり、スナップショットボリュームを再同期化することにより）破棄できます。

スナップショット作成後の作業

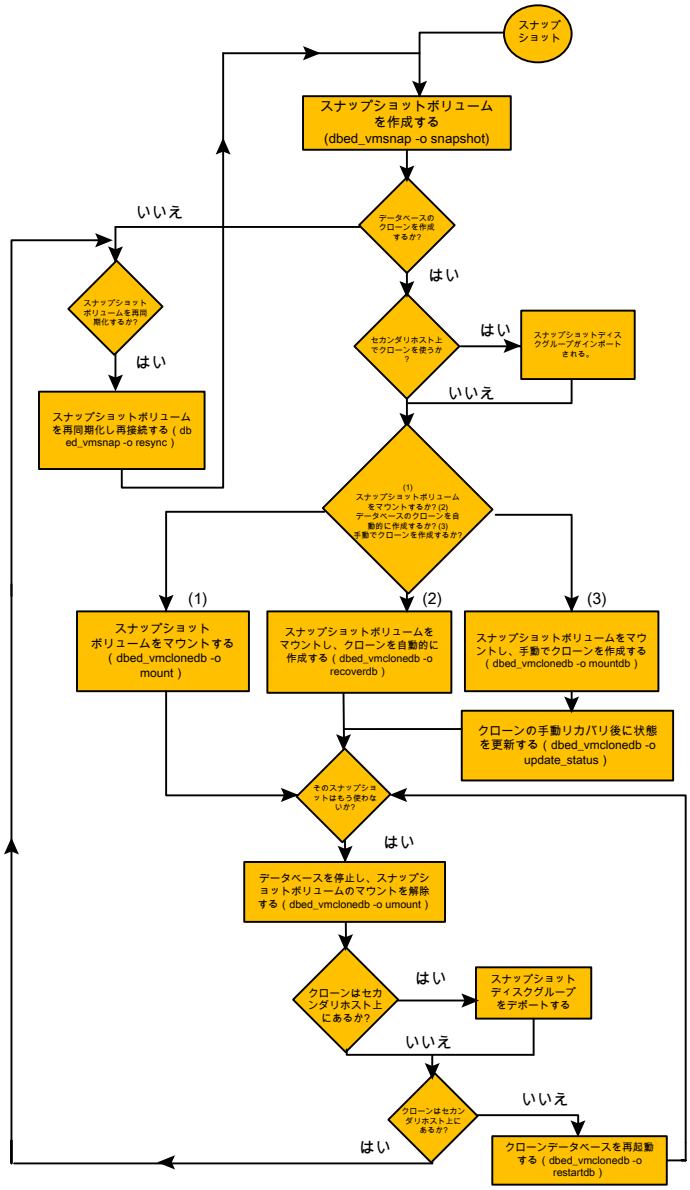
Database FlashSnap を使ってデータベースのスナップショットを作成した後は、多くの処理を実行できるようになります。バックアップとオフホスト処理を行うために、データベースのクローンを作成できます。スナップショットボリュームをプライマリデータベースと再同期化できます。

単一インスタンス Oracle では、プライマリデータベースに障害が発生した場合はスナップショットボリュームを逆再同期化することによってリカバリできます。

次のフローチャートでは、Database FlashSnap を使ってデータベースのスナップショットを作成した後に実行できる処理を示すとともに、次の 4 つの質問が含まれています。

図 32-2 では、データベースのスナップショットを作成した後に Database FlashSnap を使って実行できる処理を示しています。

図 **32-2** データベースのスナップショット作成後に実行できる処理

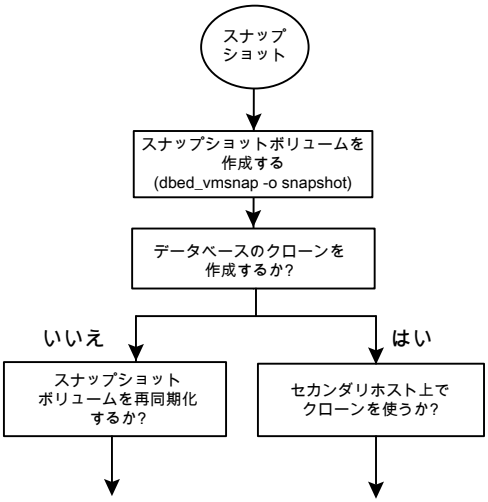


データベースのクローンを作成する場合

スナップショットのボリュームを作成した後で、データベースのクローンの作成、セカンダリホストでのクローンの使用、またはスナップショットのボリュームの再同期化を続行できます。

図 32-3 は、このプロセスのフローチャートを示します。

図 32-3 データベースのクローンの作成

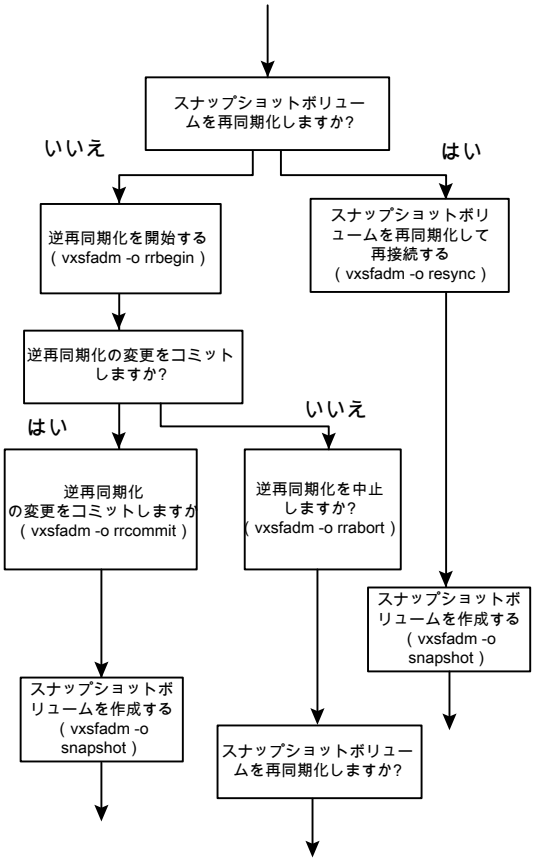


スナップショットボリュームを再同期化する場合

データベースのクローンを作成しない予定の場合、スナップショットボリュームの再同期化および再接続を続行するか、または逆再同期処理を開始できます。

図 32-4 は、このプロセスのフローチャートを示します。

図 32-4 データベースの再同期化



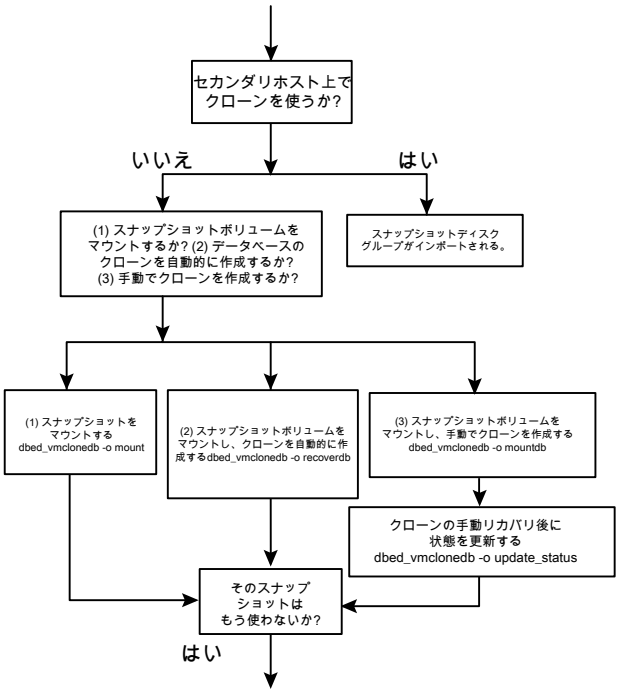
セカンダリホスト上でクローンを使用する場合

データベースのクローンを作成し、セカンダリホスト上でクローンを使う予定の場合、スナップショットディスクグループがインポートされます。それ以外の場合は、次の 3 つの選択肢のいずれかを続行します。

- スナップショットボリュームをマウントする
- スナップショットボリュームをマウントし、クローンを自動的に作成する
- スナップショットボリュームをマウントし、手動でクローンを作成する

「図 32-5」は、このプロセスのフローチャートを示します。

図 32-5 セカンダリホスト上のクローン

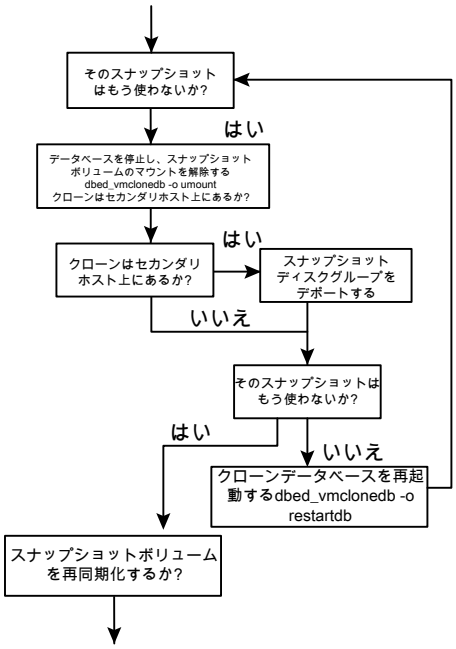


スナップショットでの作業が完了した場合

そのスナップショットをもう使わない場合、データベースを停止し、スナップショットボリュームのマウントを解除します。クローンがセカンダリホスト上にある場合、スナップショットディスクグループをデポートします。

図 32-6 は、このプロセスのフローチャートを示します。

図 32-6 スナップショットの終了



FlashSnap コマンド

次の各項で説明する FlashSnap コマンドを使うことにより、Database FlashSnap の処理を行えます。

- 「スナッププランの作成（dbed_vmchecksnap）」
- 「スナッププランの検証（dbed_vmchecksnap）」
- 「スナッププランの表示、コピー、削除（dbed_vmchecksnap）」
- 「スナップショットの作成（dbed_vmsnap）」
- 「スナップショットボリュームを使ったデータベースのバックアップ（dbed_vmclonedb）」
- 「クローンデータベースの作成（dbed_vmclonedb）」
- 「スナップショットとデータベースの再同期」
- 「スナップショットボリュームの削除」

スナッププランの作成（dbed_vmchecksnap）

dbed_vmchecksnap コマンドでは、dbed_vmsnap で Oracle データベースのスナップショット作成に使うスナッププランを作成します。

スナッププランは、スナップショットのシナリオ（online、offline、instant）を指定します。

どのスナッププランのファイルでも選んだファイルに名前を付けることができます。スナッププラン内の各エントリは、parameter=argument という形式の行になっています。

表 32-2 は、dbed_vmchecksnap コマンドを使ってスナッププランを作成または検証する場合に設定可能なパラメータをまとめたものです。

表 32-2　dbed_vmchecksnap パラメータの値

パラメータ	値
SNAPSHOT_VERSION	メジャーリリースの SF Enterprise 製品 に対応するスナップショットのバージョンを指定します。
PRIMARY_HOST	プライマリデータベースが配置されているホストの名前
SECONDARY_HOST	データベースがインポートされるホストの名前
PRIMARY_DG	プライマリデータベースによって使われる、コロンで名前が区切られた VxVM ディスクグループのリスト
SNAPSHOT_DG_PREFIX	ディスクグループ名に結合する接頭辞の名前。スナップショットディスクグループの名前は、SNAPSHOT_DG_PREFIX と、それに対応するプライマリディスクグループの名前の連結です。デフォルト値は「SNAP_」です。 スナップショットボリュームは、プライマリホスト上のこのディスクグループに配置されるとともにデポートされます。セカンダリホストでは、このディスクグループをインポートしてクローンデータベースを開始します。 SNAPSHOT_DG はリリース 5.1 ではサポートされていません。これは SNAPSHOT_DG_PREFIX に置き換わります。
ORACLE_SID	Oracle データベースの名前。デフォルトでは、スナッププランに Oracle データベースの名前が含まれています。

パラメータ	値
ARCHIVELOG_DEST	アーカイブログの絶対パス。 アーカイブログを多重化する場合にデータベースリカバリで使うことができるさまざまなアーカイブログの保存場所が用意されています。どのアーカイブログの保存場所を使うかを指定する必要があります。 SNAPSHOT_ARCHIVE_LOG が yes の場合は、別のボリュームにアーカイブログの保存場所を設けることをお勧めします。
SNAPSHOT_ARCHIVE_LOG	yes または no アーカイブログボリュームのスナップショットを作成するかどうかを指定します。アーカイブログボリュームミラーを分割してセカンダリホストにデポートする場合は、yes を指定します。Oracle リモートアーカイブログの保存先指定機能を使ってアーカイブログをセカンダリホストに送る場合、no を指定すると領域をある程度節約できます。 アーカイブログが確実にセカンダリホストに作成できるように、yes を指定するようお勧めします。
SNAPSHOT_MODE	データベースのスナップショットモードを指定します。値は online、offline、instant のいずれかです。 ■ データベースがオンラインの状態の間にスナップショットが作成された場合は、`dbed_vmsnap` コマンドを使って表領域をバックアップモードに設定します。`dbed_vmsnap` は、スナップショットの作成を終了させると、表領域のバックアップモードを解除して、余分な REDO ログがアーカイブ化されるようにログファイルを切り替え、スナップショットのアーカイブログを作成します。 ■ データベースがオフラインの場合は、表領域をバックアップモードに切り替える必要はありません。データベースはオフラインスナップショットを作成する前に停止している必要があります。 ■ データベーススナップショットが instant の場合は、`dbed_vmsnap` は、表領域をバックアップモードに変える操作を省略します。 **メモ:** SNAPSHOT_MODE が offline または instant に設定されている場合は、オフホストの設定が必要になり、`-r` *`relocate_path`* オプションは使えません。

パラメータ	値
SNAPSHOT_PLAN_FOR	デフォルト値は database であり、この値を変更することはできません。 スナップショットを作成するデータベースオブジェクトを指定します。
SNAPSHOT_PLEX_TAG	スナップショットプレックスタグを指定します。この変数を使って、スナップショットプレックスのタグを指定します。plex_tag の最大長は、15 文字です。デフォルトのプレックスタグは、dbed_flashsnap です。
SNAPSHOT_VOL_PREFIX	スナップショットボリュームの接頭辞を指定します。この変数を使って、プライマリディスクグループを分割したスナップショットボリュームの接頭辞を指定します。ボリューム名は、32 文字より長くできません。接頭辞を割り当てる場合は、ボリューム名の長さを考慮する必要があります。
SNAPSHOT_MIRROR	スナップショットとする接頭辞の数を指定します。 デフォルト値は 1 です。
DG:*VOL*	オプション -o setdefaults -m オプションを指定して dbed_vmchecksnap コマンドを実行すると、これらのエントリはデフォルトで作成されます。これらのエントリは、dbed_vmclondb を使ってクローン操作を実行するときに、関連するスナップショットボリュームのマウントパスを指定します。これらのフィールドの値は空白のままにすることができます。これらが空白の場合は、dbed_vmclonedb 操作を実行するときに -r relocate_path を指定する必要があります。

はじめに dbed_vmchecksnap を実行する場合は、-o setdefaults オプションを使って、変数にデフォルト値を使うスナッププランを作成します。このとき、別のスナップショットシナリオの変数を設定する場合にファイルを手動で編集してもかまいません。

スナッププランを作成する前に、次の条件を満たしているかどうかを確認します。

前提条件

- ストレージは、次の指定どおりに設定してください。
  p.188 の「Database FlashSnap のホストとストレージの準備」を参照してください。
- Oracle データベース管理者である必要があります。
- ディスクグループは、バージョン 110 以上としてください。ディスクグループのバージョンについて詳しくは、vxdg(1M)のマニュアルページを参照してください。
- DCO と DCO ボリュームが、スナップショットを作成するボリュームと関連付けられていることを確認してください。
- スナップショットプレックスとそれに関連する DCO ログは、元のプレックスとは別のディスク上に配置し、システム管理者がスナップショットを作成できるように正しく設定を行う必要があります。
- 永続 FastResync は、既存のデータベースボリューム上で有効にし、ディスクは、スナップショットボリュームに割り当てる必要があります。
- データベースは、アーカイブログモードで稼動している必要があります。アーカイブログモードは、Oracle 初期化パラメータファイルに設定されています。
- Oracle データベースは、必須アーカイブの記録先を少なくとも 1 つ保持している必要があります。
- ORACLE_HOME は、スナップショットに使うディスク上には配置できません。
- Oracle データベースファイルとアーカイブログファイルは、同じディスクグループの中にある一意のディスクを使う異なるボリュームを使う必要があります。

使用に関する注意事項

- オフホスト処理の実行中は、スナッププランをプライマリホスト上に作成してください。
- dbed_vmchecksnap コマンドを使ってスナッププランを作成した後は、必要に応じてテキストエディタを使ってこのファイルを確認したり更新したりできます。
- 作成したスナッププランを保存するローカルの作業ディレクトリを作成することをお勧めします。
- 詳しくは、dbed_vmchecksnap(1M)のオンラインマニュアルページを参照してください。
- データベースの SNAPSHOT_MODE が online に設定されている場合、プライマリホストがセカンダリホストと同じホストである可能性があります。
- SNAPSHOT_MODE を offline または instant に設定する場合は、プライマリホストとセカンダリホストを別にする必要があります。

スナッププランを作成するには

1 Oracle データベース管理者として次の手順を実行します。

2 作成したスナッププランを保存する作業ディレクトリにディレクトリを変更します。

```
$ cd /working_directory
```

3 dbed_vmchecksnap コマンドを使ってデフォルト値でスナッププランを次のように作成します。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S  ORACLE_SID ¥
-H ORACLE_HOME -f SNAPPLAN -o setdefaults -t  host_name ¥
[-p PLEX_TAG] [-m]
```

multi-dg を使い、マップされたマウントポイントに -m オプションを指定した場合の出力例:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S slave -H $ORACLE_HOME ¥
-f snapplan3 -o setdefaults -t mensa -m
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap will
be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snapplan3 for slave
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=mensa
SECONDARY_HOST=mensa
PRIMARY_DG=datadg1:datadg2:archdg1
ORACLE_SID=slave
ARCHIVELOG_DEST=/arch/sfae10g
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
archdg1:arch1vol1=
datadg1:data1vol2=
datadg1:data1vol1=
datadg2:data1vol1=
datadg2:data1vol2=
```

4 テキストエディタでスナッププランファイルを開き、必要に応じて修正します。

スナップショットイメージに対して作成するスナッププラン例

この例では、1 ノード構成のスナップショットイメージに対するスナッププラン snap1 を作成し、デフォルト値を設定します。このホストには、host1 という名前が付けられ、作業ディレクトリは /export/snap_dir です。

次は dbed_vmchecksnap コマンドと出力の例です。

```
$ cd /export/snap_dir

$ $ /opt/VRTS/bin/dbed_vmchecksnap -S PROD ¥
-H /oracle/product/orahome -f snap1 -o setdefaults -t host1
```

単一ディスクグループ環境の出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap will be deprecated
in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername use
/opt/VRTS/bin/vxsfadm
Snapplan snap1 for PROD
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host1
PRIMARY_DG=proddg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch/sfae10g
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

複数ディスクグループ環境の出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snap1 for PROD
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=proddg1:proddg2
ORACLE_SID=PROD
```

```
ARCHIVELOG_DEST=/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=my_tag
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

この 2 つ目の例では、クラスタ設定内の 2 ノードにスナップショットイメージに対するスナッププラン snap 2 を作成し、デフォルト値を設定します。プライマリホストは host1、セカンダリホストは host2、作業ディレクトリは /export/snap_dir です。

次は dbed_vmchecksnap コマンドと出力の例です。

```
$cd /export/snap_dir

$/opt/VRTS/bin/dbed_vmchecksnap -S PROD ¥
-H /oracle/product/orahome -f snap2 -o setdefaults -t host2
```

単一ディスクグループ環境の出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snap2 for PROD
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=proddg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch/sfae10g
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

複数ディスクグループ環境の出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snap2 for PROD
```

```
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=proddg1:proddg2
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch/sfae10g
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

デフォルトでは、スナッププランの SNAPSHOT_PLEX_TAG 値は dbed_flashsnap として設定されます。別のタグ名を割り当てるには、-p オプションを使います。setdefaults オプションでスナッププランを作成する場合は、-p オプションを利用します。

次の例では、SNAPSHOT_PLEX_TAG 値として *my_tag* を割り当てるのに、setdefaults とともに -p オプションが使われています。

```
$ dbed_vmchecksnap -S PROD -H $ORACLE_HOME -o setdefaults ¥
-p my_tag -f snap1 -t host2
```

単一ディスクグループ環境の出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snap1 for PROD
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=proddg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=my_tag
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

複数ディスクグループ環境の出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snap1 for PROD
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=proddg1:proddg2
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=my_tag
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

マップ済みマウントオプションを有効にしたスナッププランの作成例

マップ済みマウントオプションを有効にしたスナッププランの作成例です。

```
$/opt/VRTS/bin/dbed_vmchecksnap -S PROD ¥
   -H /oracle/product/orahome -f snap2 -o setdefaults -t host2 -m
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snap2 for PROD
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=host1
SECONDARY_HOST=host2
PRIMARY_DG=proddg
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
```

```
PRODdg:data_vol=/tmp/datadst
PRODdg:arch_vol=/tmp/archdst
```

dbed_vmchecksnap で -m オプションを使った場合の環境の出力例:

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S slave -H $ORACLE_HOME ¥
-f snapplan3 -o setdefaults -t mensa -m

SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Snapplan snapplan3 for slave
SNAPSHOT_VERSION=5.0
PRIMARY_HOST=mensa
SECONDARY_HOST=mensa
PRIMARY_DG=datadg1:datadg2:archdg1
ORACLE_SID=PROD
ARCHIVELOG_DEST=/arch
SNAPSHOT_ARCHIVE_LOG=yes
SNAPSHOT_MODE=online
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
archdg1:arch1vol1=
datadg1:data1vol2=
datadg1:data1vol1=
datadg2:data1vol1=
datadg2:data1vol2=
```

スナッププランを検証する前に、スナッププランを編集し、マウントポイントを *dg:volume* ペアに追加します。入力したマウントポイントは、dbed_vmclonedb 操作後にクローンデータベースがマウントされる場所になります。

マルチミラースナップショットの作成

データベーススナップショットの可用性を高めるには、スナップされたスナップショットボリュームに、2 つ以上のミラーを含めてください。これにより、たとえミラーのうち 1 つが無効となったとしても、スナップショットボリュームは使えるようになります。ミラーのいずれかが無効になった場合でも、スナップショットボリュームをマウントしてデータベーススナップショット全体を使うことができます。マルチミラースナップショットは、スナッププランの SNAPSHOT_MIRROR=<n> で有効になります。

メモ: Flashsnap ツールのコマンドラインの書式や引数に特に変更はありません。

スナップショットを作成する前に、タグが付けられたすべてのスナップショットミラーの状態が SNAPDONE であることを確認します。

スナップショットミラーについて詳しくは、『Storage Foundation 管理者ガイド』を参照してください。

スナッププランの検証（dbed_vmchecksnap）

スナッププランを作成した後は、次の手順により、スナッププランのパラメータを検証し、スナップショットの作成に対してスナップショットボリュームが正しく設定されているかどうかをチェックします。検証がうまくいった場合は、スナッププランがリポジトリにコピーされます。`dbed_vmchecksnap` コマンドに `-o validate` オプションを指定してスナッププランを検証します。

スナッププランを検証する前に、次の前提条件と留意点を確認します。

前提条件

- `dbed_vmchecksnap` コマンドが実行している間に、データベースが起動し、起動した状態である必要があります。

使用に関する注意事項

- `dbed_vmchecksnap` コマンドは、Oracle データベース管理者が実行する必要があります。
- AIX の場合:
  デフォルト値では、検証を強制します。検証をスキップする場合は -n オプションを使用します。
- `dbed_vmchecksnap -o validate` を使って、スナッププランとストレージを検証する場合は、検証結果を保存できます。システム管理者は、検証に失敗した場合にこの検証結果を使ってストレージの設定を調整できます。
- スナッププランが更新または修正された場合は、そのスナッププランを再検証する必要があります。データベースのディスクグループに変更が加えられた場合は、スナッププランを再度検証することをお勧めします。
- 詳しくは `dbed_vmchecksnap`（1M）のマニュアルページを参照してください。

メモ: 次の手順では、Oracle データベース管理者としてコマンドを実行する必要があります。

スナッププランを検証するには

1 お手持ちのスナッププランが保存されている場所をディレクトリから作業ディレクトリに変更するには、次のように入力します。

```
$ cd /working_directory
```

2 dbed_vmchecksnap コマンドを使ってスナッププランを検証するには、次のように入力します。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S  ORACLE_SID ¥
-H ORACLE_HOME -f SNAPPLAN -o validate
```

スナップショットイメージに対するスナッププラン snap1 の検証例

次の例では、同一ノード構成のスナップショットイメージに対するスナッププラン snap1 を検証します。プライマリホストは host1 であり、作業ディレクトリは /export/snap_dir です。コマンドは Oracle データベース管理者として発行されます。

次は dbed_vmchecksnap コマンドと出力の例です。

```
$ cd /export/snap_dir
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -H /oracle/product/orahome
 ¥
-f snap1 -o validate
```

出力例:

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Validating database configuration for third-mirror-break-off snapshot:
ORACLE_SID = PROD
ORACLE_HOME = /oracle/product/orahome
APP_MODE = online
SNAPSHOT_ARCHIVE_LOG = yes
REVERSE_RESYNC_ARCHIVE_LOG = no
REVERSE_RESYNC_RECOVERY = auto
ARCHIVELOG_DEST = /arch
Database validation successful.
Validating database volume layout for third-mirror-break-off snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup datadg mounted on /data.
Archivelog volume ready for snapshot:
```

```
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
Storage units to be used for snapshot from diskgroup datadg:
hitachi_usp-vm0_01c4 hitachi_usp-vm0_01c6
SNAPSHOT_VOL_PREFIX = SNAP_
SNAPSHOT_DG_PREFIX = SNAP_
Database volume layout validated successfully.
```

スナッププランの表示、コピー、削除（dbed_vmchecksnap）

指定された Oracle データベースに含まれるすべてのスナッププランの一覧表示、スナッププランファイルの表示、スナッププランのコピーや削除などを行うときは、次の使用に関する注意事項を考慮してください。

使用に関する注意事項

- ローカルスナッププランを更新または変更した場合は、再検証する必要があります。
- データベーススキーマまたはディスクグループを変更した場合は、dbed_update を実行した後で、再検証する必要があります。

スナッププランの表示

dbed_vmchecksnap コマンドを使って利用可能なすべてのスナッププランを一覧表示し、特定のスナッププランの詳しい情報を表示できます。

特定の Oracle データベースの使用可能なスナッププランをすべて表示するには

◆ 次のように dbed_vmchecksnap コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -o list
```

次の例では、利用可能なすべてのスナッププランがデータベース PROD 用としてリストされます。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -o list
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an alternname
use /opt/VRTS/bin/vxsfadm
The following snapplan(s) are available for sfae10g:
SNAP_PLAN  SNAP_STATUS  DB_STATUS  SNAP_READY
snap1  init_full  -   yes
snap2  init_full  -   yes
```

コマンド出力には、利用可能なすべてのスナップショット、それらのスナップショットの状態（SNAP_STATUS）、データベースの状態（DB_STATUS)、スナップショットを作成できるかどうか（SNAP_READY）が表示されます。

Database FlashSnap の状態については、

p.481 の「Database FlashSnap の状態情報」を参照してください。

スナッププランの詳細情報を表示するには

◆ 次のように dbed_vmchecksnap コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ¥
ORACLE_SID -f SNAPPLAN -o list
```

次の例では、スナッププラン snap1 が表示されます。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o list

SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an alternname
use /opt/VRTS/bin/vxsfadm
SNAPSHOT_VERSION=5.0
ORACLE_SID=sfaedb1
ORACLE_HOME=/ora_base/db_home
SNAPSHOT_PLEX_TAG=dbed_flashsnap
SNAPSHOT_DG_PREFIX=SNAP_
SNAPSHOT_VOL_PREFIX=SNAP_
SNAPSHOT_MIRROR=1
SNAPSHOT_MODE=online
SNAPSHOT_ARCHIVE_LOG=yes

STORAGE_INFO
sdu
sdx
SNAP_PLEX=datavol-02 archvol-02 datavol_dcl-02 archvol_dcl-02

STATUS_INFO
SNAP_STATUS=init_full
```

スナッププランのコピー

既存のスナッププランに類似したスナッププランを作成する場合は、既存のスナッププランのコピーを作成し、それを修正します。SFDB リポジトリから現在のディレクトリへスナッププランをコピーするには、現在のディレクトリにそのスナッププランが存在していないことが必要です。

SFDB リポジトリから現在のディレクトリにスナッププランをコピーします。

◆ 次のように dbed_vmchecksnap コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S  ORACLE_SID ¥
-f SNAPPLAN -o copy
```

次の例では、VxDBA リポジトリから現在のディレクトリにスナッププラン snap1 がコピーされます。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snap1 -o  copy
SFDB vxsfadm WARNING V-81-0000 dbed_vmchecksnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername use
/opt/VRTS/bin/vxsfadm
Copying 'snap1' to '/export/snap_dir'
```

スナッププランの削除

不要となったスナッププランは、ローカルのディレクトリやリポジトリから削除できます。

スナッププランを SFDB リポジトリから削除するには

◆ 次のように dbed_vmchecksnap コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmchecksnap -S ORACLE_SID -f¥
SNAPPLAN -o remove
```

スナップショットの作成（dbed_vmsnap）

dbed_vmsnap コマンドでは、データベースで使用するミラーボリュームをスナップショットデータベースに分割することにより、Oracle データベースのスナップショットを作成します。スナップショットイメージは、データベースと同じホスト上か、またはセカンダリホスト上（ストレージをプライマリホストと共有している場合）のいずれかで使うことができます。

dbed_vmsnap によって作成されるスナップショットイメージは、Oracle データベースのデータファイルの静止イメージです。dbed_vmsnap コマンドでは、スナップショットデータベースの作成時にバックアップ制御ファイルが必ず作成されます。このファイルを使うと、必要に応じてデータの完全な回復が行えます。

Database FlashSnap の状態については、

p.481 の「Database FlashSnap の状態情報」を参照してください。

前提条件

- Oracle データベース管理者としてログインする必要があります。
- dbed_vmsnap でスナップショットイメージを作成する前に、dbed_vmchecksnap を使ってスナッププランの作成と検証を行っておく必要があります。

使用に関する注意事項

- オフホスト処理の実行中は、プライマリホスト上で dbed_vmsnap コマンドを使用してください。
- Oracle データベースファイルとその他のソフトウェアの間でボリュームを共有しないでください。
- スナップショットボリュームを作成するときは、プライマリボリュームの別のコントローラやディスク上にスナップショットを作成します。
- Oracle データベースボリュームとは別の場所にアーカイブログが保存されるようにしてください。
- $ORACLE_HOME/dbs ディレクトリに、制御ファイルを含むいかなるデータファイルも格納しないでください。
- ミラーがブレークオフしている場合は、プライマリとスナップショットボリュームで変更されたデータ量に応じて再同期処理の速度が変化します。
- 詳しくは dbed_vmsnap（1M）のマニュアルページを参照してください。

次の点に注意してください。

- スナップショットを強制的に作成するには、-F オプションを使います。-F オプションを使うには、操作に失敗したスナップショットの問題が SFDB コマンドを使わないで解消されている（つまり、ボリュームは VxVM コマンドを使って同期化されている）必要があります。このような場合は、スナッププランをスナップショットの作成に利用できないという状態が表示されます。-F オプションでは、作成不能の状態を無視して、ボリュームの可用性をチェックします。ボリュームがこのチェックに合格すると、スナップショットを作成します。
- スナップショットの作成後、dbed_vmsnap により、dbed_vmclonedb の実行に必要な値が返ります。これらの値には、スナップショットディスクグループ、スナッププラン名、オフホスト構成用の SFDB リポジトリボリュームなどが含まれます。これらの値を記録し、dbed_vmclonedb の実行時に使います。
- また、dbed_vmchecksnap -f snapplan -o list コマンドを使って、スナップショットディスクグループ、スナッププラン名、SFDB リポジトリに関する情報を取得することもできます。

メモ: 次の手順では、Oracle データベース管理者としてコマンドを実行する必要があります。

スナップショットを作成するには

1 次のように入力して、お手持ちのスナッププランが保存されている作業ディレクトリに変更します。

```
$ cd /working_directory
```

2 スナッププランの SNAPSHOT_MODE が offline に設定されている場合は、データベースを停止します。

3 次の dbed_vmsnap コマンドを使ってスナップショットイメージを作成します。

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f  SNAPPLAN ¥
-o snapshot

$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN ¥
-o snapshot

$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN ¥
-o snapshot [-F]
```

スナップショットボリュームは、この時点で、整合性のとれたデータベースのバックアップコピーとなっています。このスナップショットボリュームをテープやその他のバックアップメディアにコピーすれば、データベースをバックアップできます。

p.422 の「スナップショットボリュームを使ったデータベースのバックアップ（dbed_vmclonedb）」を参照してください。

4 また、意思決定支援のために Oracle データベースをもう 1 つ作成することもできます。

p.426 の「クローンデータベースの作成（dbed_vmclonedb）」を参照してください。

データベースのスナップショットイメージ PROD の作成例

この例では、同一ノード構成におけるデータベースのスナップショットイメージ PROD を作成します。ここでは、SECONDARY_HOST パラメータに、スナッププランの PRIMARY_HOST パラメータと同じ設定値が割り当てられています。

メモ: 次の手順では、Oracle データベース管理者としてコマンドを実行する必要があります。

```
$ /opt/VRTS/bin/dbed_vmsnap -S PROD -f snap1 -o snapshot

SFDB vxsfadm WARNING V-81-0000 dbed_vmsnap
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
```

```
use /opt/VRTS/bin/vxsfadm
Validating database configuration for third-mirror-break-off snapshot:
ORACLE_SID = PROD
ORACLE_HOME = /ora_base/db_home
APP_MODE = online
ARCHIVELOG_DEST = /arch
SNAPSHOT_ARCHIVE_LOG = yes
REVERSE_RESYNC_ARCHIVE_LOG = no
REVERSE_RESYNC_RECOVERY = auto
Database validation successful.
snapshot started at Mon Aug 19 05:14:12 2013.
Putting database in backup mode ... Done
Validating database volume layout for third-mirror-break-off snapshot:
Data volumes ready for snapshot:
Volume/volume-set datavol of diskgroup datadg mounted on /data.
Archivelog volume ready for snapshot:
Volume/volume-set archvol of diskgroup datadg mounted on /arch.
Storage units to be used for snapshot from diskgroup datadg:
hitachi_usp-vm0_01c4 hitachi_usp-vm0_01c6
SNAPSHOT_VOL_PREFIX = SNAP_
SNAPSHOT_DG_PREFIX = SNAP_
Database volume layout validated successfully.
Creating snapshot volumes for data volumes ... Done
Taking database out of backup mode ... Done
Flushing online redo logs ... Done
Creating snapshot volume for archivelog volume ... Done
Copying snapshot information to snapshot volume ... Done
Creating snapshot diskgroups ... Done
Deporting snapshot diskgroups ... Done
SNAP_datadg
snapshot ended at Mon Aug 19 05:14:53 2013.
```

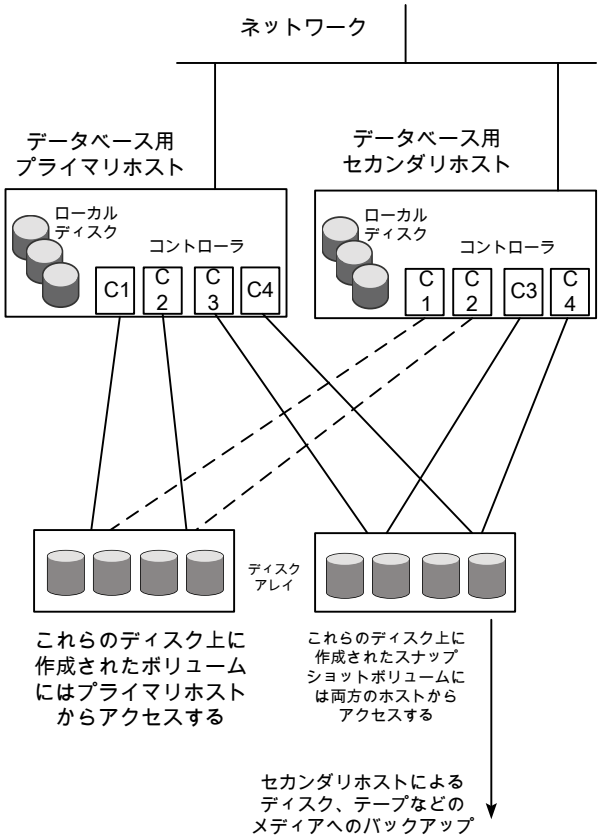
スナップショットボリュームを使ったデータベースのバックアップ（dbed_vmclonedb）

データベースのバックアップで最も一般的に使われるバックアップ元はスナップショットです。スナップショットボリュームを使うことの利点は、バックアップの実行で物理デバイスの I/O 帯域幅と競合しないことにあります。スナップショットボリュームをセカンダリホスト上で使えるようにすると、プライマリホスト上でのバックアップにおいて CPU や I/O アダプタに余分な負荷がかからなくなります。

また、クローンデータベースは、プライマリデータベースのバックアップとしても有効に機能します。スナップショットボリュームを使うことで、プライマリデータベースをテープにバックアップできます。

図 32-7 は、スナップショットボリュームがセカンダリホスト上にある場合の一般的な設定を示しています。

図 32-7　セカンダリホストでデータベースのバックアップを行う場合のシステム構成例



前提条件

- dbed_vmclonedb コマンドを使うには、Oracle データベース管理者としてログインする必要があります。
- dbed_vmclonedb コマンドを使うには、事前にスナッププランの検証、さらにスナップショットの作成が必要です。
  p.422 の「スナップショットボリュームを使ったデータベースのバックアップ（dbed_vmclonedb）」を参照してください。
  p.414 の「スナッププランの検証（dbed_vmchecksnap）」を参照してください。
  p.419 の「スナップショットの作成（dbed_vmsnap）」を参照してください。
- ボリュームスナップショットにデータベース全体を含む必要があります。
- スナップショットイメージの初期マウントポイントを指定する -r *relocate_path* オプションを指定して dbed_vmclonedb コマンドを使うには、事前にシステム管理者がマウントポイントを作成し、その所有者を Oracle データベース管理者に変更しておく必要があります。

使用に関する注意事項

- dbed_vmclonedb コマンドは、セカンダリホストで使えます。
- 同一ノード構成の場合は、プライマリホストとセカンダリホストは同一です。
- 同一ノード構成で、マップ済みのマウントが指定されていなかったか、またはマップ済みのマウントエントリの一部がスナッププランで空白のままになっていた場合は、-r *relocate_path* が必要です。
- 詳しくは dbed_vmclonedb（1M）のマニュアルページを参照してください。

スナップショットボリュームのマウントとバックアップ

スナップショットボリュームを使ってバックアップを実行する前に、まずそのスナップショットボリュームをマウントする必要があります。

メモ: 次の手順では、Oracle データベース管理者としてコマンドを実行する必要があります。

メモ: Oracle のオンラインバックアップ方法を適用する場合は、アーカイブログファイルについてもすべてバックアップを実行する必要があります。これは、データベースの完全なリストアやリカバリを行えるようにするためです。

スナップショットボリュームをマウントするには

◆ 次のように dbed_vmclonedb コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID ¥
  -o mount,new_sid=new_sid,server_name=svr_name¥
  -f SNAPPLAN [-H ORACLE_HOME] ¥
  [-r relocate_path]
```

これで、特定のディレクトリ下にある個々のファイルやファイルグループをメディアにバックアップできるようになりました。

この例では、スナップショットボリュームがマウントされています。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S PROD ¥
 -o mount,new_sid=NEWPROD,server_name=svr_name ¥
 -f snap1 -r /clone/single

SFDB vxsfadm WARNING V-81-0000 dbed_vmclonedb
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an alternname
use /opt/VRTS/bin/vxsfadm
Retrieving snapshot information ... Done
Importing snapshot diskgroups ... Done
Mounting snapshot volumes ... Done
```

メモ: オフホスト設定用として上記のコマンドに server_name を指定しないと、書式エラーが表示されます。

スナップショットボリュームから継承した **Storage Checkpoint** をセカンダリホストにマウントするには

1 セカンダリホスト上で、プライマリデータベースから継承した Storage Checkpoint を dbed_ckptdisplay コマンドを使って一覧で表示します。

次に例を示します。

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S ORACLE_SID -n
```

2 一覧で表示された Storage Checkpoint のいずれかを dbed_ckptmount コマンドを使ってマウントできます。

次に例を示します。

```
$ /opt/VRTS/bin/dbed_ckptmount -S ORACLE_SID -c CKPT_NAME ¥
  -m MOUNT_POINT
```

次の制限があります。

- 次のコマンドを実行する場合は、事前に、マウント中のすべての Storage Checkpoint をマウント解除しておく必要があります。

```
$ /opt/VRTS/bin/dbed_ckptumount -S ORACLE_SID -c CKPT_NAME
```

- `-o mount` オプションを指定し、`-r relocate_path` を使用せずに `dbed_vmclonedb` コマンドでスナップショットボリュームをマウントした場合、オフホスト設定でスナップショットボリュームとともに継承した Storage Checkpoint しかマウントできません。
- `dbed_vmclonedb` コマンドに `-o mount` オプションを指定してクローンデータベースを作成する前に、スナップショットボリュームから継承した Storage Checkpoint をマウントできます。ただし、`dbed_vmclonedb` コマンドに `-o recoverdb` オプションを指定してクローンデータベースを作成すると、Storage Checkpoint は消失します。

スナップショットを使ってデータベースのバックアップを作成するには

◆ スナップショットボリュームを、テープやその他の適切なバックアップメディアにコピーします。

バックアップからの復元

単一インスタンスの Oracle 環境では、ディスク障害によって失われたボリュームまたは人的エラーによって破壊されたデータを復元するのにバックアップコピーが使用されます。ボリュームのデータが破損しており、バックアップから復元する必要があると分かっている場合は、Database FlashSnap の逆再同期機能を使用してデータベースを復元できます。

クローンデータベースの作成（dbed_vmclonedb）

SFDB コマンドでは、スナップショットボリュームを使ってクローンデータベースを作成できます。プライマリデータベースのスナップショットを使ったクローンデータベースの作成は、前もって定められた特定時点で実施できます。その後は、意思決定支援の分析と報告を実施できます。この操作では、データをクローンデータベースから取得し、プライマリデータベースからは取得しません。このため、実稼動データベースに余分な負荷がかからなくなります。

また、クローンデータベースは、プライマリデータベースのバックアップとしても有効に機能します。

p.422 の「スナップショットボリュームを使ったデータベースのバックアップ（dbed_vmclonedb）」を参照してください。

また、スナップショットボリュームを使うことで、プライマリデータベースのバックアップをテープに作成することもできます。

Database FlashSnap の再同期機能により、プライマリデータベースから取得した最新情報を使ってクローンデータベースをすばやく更新できます。意思決定支援データの更新に要する時間を短縮することも、分析報告の生成頻度を高めることにつながります。

Database FlashSnap によるデータベースのクローンの作成

同一ノード構成の場合は、dbed_vmclonedb コマンドを使うと、同一のホスト上でクローンデータベースが作成されます。また、このコマンドを使って、作成したクローンデータベースをシャットダウンし、そのファイルシステムをマウント解除することもできます。同一ノード構成でクローンデータベースの作成やマウント解除を行う場合は、クローンデータベースのファイルシステムがプライマリデータベースによって使われているマウントポイント以外のマウントポイントを使うように、-r *relocate_path* を指定する必要があります。

オフホスト構成では、dbed_vmclonedb コマンドは、スナップショットのディスクグループをインポートし、スナップショットをマウントし、クローンデータベースを起動します。また、このコマンドを使って前述の手順を逆から実行させることもできます。その場合は、クローンデータベースのシャットダウン、ファイルシステムのマウント解除、スナップショットディスクグループのデポートの順に行います。

警告: クローンデータベースを作成する場合は、元のデータベースに含まれるすべての Storage Checkpoint が破棄されます。

前提条件

- Oracle データベース管理者としてログインする必要があります。
- dbed_vmclonedb コマンドを使うには、事前にスナッププランの検証、さらにスナップショットの作成が必要になります。
  p.394 の「データベーススナップショットの作成について」を参照してください。
  p.414 の「スナッププランの検証（dbed_vmchecksnap）」を参照してください。
  p.419 の「スナップショットの作成（dbed_vmsnap）」を参照してください。
- ボリュームスナップショットにデータベース全体を含む必要があります。
- システム管理者は、データベース管理者に対し、必要なボリュームとマウントポイントへのアクセスを許可する必要があります。
- スナップショットイメージの初期マウントポイントを指定する -r *relocate_path* オプションを指定して dbed_vmclonedb コマンドを使うには、事前にシステム管理者がマウントポイントを作成し、その所有者を Oracle データベース管理者に変更しておく必要があります。
- Oracle データベースは、必須アーカイブの記録先を少なくとも 1 つ保持している必要があります。
- マップ済みのマウントの場合には、システム管理者がマップ済みのマウントポイントを作成する必要があります。

使用に関する注意事項

- dbed_vmclonedb コマンドは、セカンダリホストで使えます。
- 同一ノード構成で、マップ済みのマウントが指定されていなかったか、またはマップ済みのマウントエントリの一部がスナッププランで空白のままになっていた場合は、-r relocate_path が必要です。
- クローンデータベースの初期化パラメータは、プライマリデータベースからコピーされます。つまり、クローンデータベースは、プライマリデータベースと同じメモリやマシンリソースを継承することになります。クローンデータベースのメモリの必要条件を少なくしたい場合は、クローンデータベースを一度シャットダウンした後、メモリの必要条件が少ない、別の init.ora ファイルを使って再起動します。dbed_vmclonedb を実行するホストにおいて、使えるメモリがほとんど残っていない場合は、クローンデータベースを起動できずにクローン操作に失敗することがあります。
- 詳しくは dbed_vmclonedb（1M）のマニュアルページを参照してください。

データベースのマウントやリカバリを手動で行うには、次の手順を実行します。

1. SFDB コマンドを使用してクローンデータベースを起動してマウントし、手動でのデータベースリカバリを許可します。
2. Oracle のリカバリ手順に従って、データベースを手動でリカバリします。
3. SFDB コマンドを使用して SFDB リポジトリのクローンデータベースのスナップショット状態情報を更新します。

例: クローンデータベースを起動せずにファイルシステムをマウントする

この例では、クローンデータベースを起動せずにファイルシステムをマウントします。クローンデータベースを使うには、事前に、クローンデータベースの作成とリカバリを手動で行う必要があります。この例は、クローンデータベースとプライマリデータベースが同じホスト上で作成されている場合を対象としたものです。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S FLAS11r2 ¥
 -o mountdb,new_sid=clone1,server_name=motmot -f snap1 ¥
 -r /cudb
```

```
dbed_vmclonedb -S PROD -H /ora_base/db_home -o mountdb,
new_sid=clons1, server_name=host1 -f snap1
SFDB vxsfadm WARNING V-81-0000 dbed_vmclonedb
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an alternname
use /opt/VRTS/bin/vxsfadm
Retrieving snapshot information ...                              Done
Importing snapshot diskgroups ...                                Done
Mounting snapshot volumes ...                                    Done
```

```
Preparing parameter file for clone database ...                Done
Mounting clone database clons1 ...                             Done
```

手動によるリカバリが完了したら、プライマリホスト上にあるクローンデータベースのステータス（database_recovered）を更新する必要があります。

```
$ /opt/VRTS/bin/dbed_vmclonedb -o update_status,¥
  new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
```

データベースのクローンを自動で作成するには

◆ 次のように dbed_vmclonedb コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID ¥
  -o recoverdb,new_sid=new_sid,server_name=svr_name -f SNAPPLAN
¥
  [-H ORACLE_HOME] [-r relocate_path]
```

ここで、

ORACLE_SID	スナップショットの作成に使う Oracle のデータベースの名前です。
new_sid	クローンデータベースの ORACLE_SID を指定します。
server_name	サーバー名を svr_name として指定します。
SNAPPLAN	スナッププランファイルの名前です。
ORACLE_HOME	ORACLE_SID データベースの ORACLE_HOME の設定です。
relocate_path	スナップショットイメージの初期マウントポイントの名前です。

セカンダリホストのデータベースのクローンを作成する場合は、スナッププランファイルの PRIMARY_HOST と SECONDARY_HOST のパラメータが異なっていることを確認します。

-o recoverdb オプションがdbed_vmclonedb とともに使われている場合は、すべての利用可能なアーカイブログを使ってクローンデータベースが自動的に回復します。-o recoverdb オプションが使われていない場合は、PIT リカバリを手動で実行できます。

次の例では、プライマリデータベースのクローンが自動的に作成されます。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S FLAS11r2 ¥
  -o recoverdb,new_sid=clone1,server_name=motmot.veritas.com¥
 -f snap1 -r /cudb
```

```
SFDB vxsfadm WARNING V-81-0000 dbed_vmclonedb
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Retrieving snapshot information ... Done
Importing snapshot diskgroups ... Done
Mounting snapshot volumes ... Done
Preparing parameter file for clone database ... Done
Mounting clone database clone1 ... Done
Starting database recovery ...
Clone database clone1 is open.
```

クローンデータベースの停止とファイルシステムのマウント解除

クローンデータベースを使う処理を行っている場合は、dbed_vmclonedb -o umount コマンドを使って、クローンデータベースのシャットダウンと、すべてのスナップショットファイルシステムのマウント解除を行うことができます。クローンデータベースが、プライマリホストとディスクを共有するセカンダリホスト上で使われている場合は、-o umount オプションにより、スナップショットディスクグループもデポートされます。

メモ: マウント中の Storage Checkpoint については、そのすべてをマウント解除してから dbed_vmclonedb -o umount を実行します。

クローンデータベースをシャットダウンし、すべてのスナップショットファイルシステムをマウント解除するには

◆ 次のように dbed_vmclonedb コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmclonedb ¥
-o umount,new_sid=NEWPROD,server_name=svr_name -f snap1 -r /clone
SFDB vxsfadm WARNING V-81-0000 dbed_vmclonedb
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername
use /opt/VRTS/bin/vxsfadm
Shutting down clone database clone1 ...
Database shut down.
Retrieving snapshot information ... Done
Unmounting snapshot volumes ... Done
Deporting snapshot diskgroups ... Done
```

クローンデータベースの再起動

dbed_vmclonedb -o umount を使ったか、またはシステムを再起動したためにクローンデータベースが停止した場合は、-o restartdb オプションを使ってデータベースを再起動できます。

メモ: このオプションは、クローンデータベースの作成が正常に行われた場合にのみ使うことができます。クローンデータベースのリカバリを手動で行った場合は、-o restartdb より先に -o update_status を実行して状態を更新します。

クローンデータベースを起動するには

◆ 次のように dbed_vmclonedb コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S ORACLE_SID ¥
  -o restartdb,new_sid=new_sid,server_name=svr_name ¥
  -f SNAPPLAN [-H ORACLE_HOME] ¥
  [-r relocate_path]
```

この例では、プライマリデータベースと同じホスト上でクローンデータベースを再起動します(同一ノード構成)。

```
$ /opt/VRTS/bin/dbed_vmclonedb -S FLAS11r2 ¥
  -o restartdb,new_sid=clone1,server_name=motmot ¥
  -f snap1 -r /cudb

SFDB vxsfadm WARNING V-81-0000 dbed_vmclonedb
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an alternname
use /opt/VRTS/bin/vxsfadm
Retrieving snapshot information ... Done
Importing snapshot diskgroups ... Done
Mounting snapshot volumes ... Done
Starting clone database clone1 ...
Clone database started.
```

Oracle 一時ファイルの再作成

一時ファイルがスナップショットボリューム上にある場合は、クローンデータベースを作成して開いたときに、一時ファイルが追加されます。データファイルと同一のファイルシステム上に一時ファイルが含まれていなかった場合、dbed_vmsnap では、スナップショットに基盤となるボリュームが含まれていません。このような状況の場合には、dbed_vmclonedb

により警告メッセージが発行されるので、次の手順に従い、必要な一時ファイルをクローンデータベース上で再作成できます。

Oracle 一時ファイルを再作成するには

1. 一時ファイルをデータファイルと同じファイルシステムに保存していない場合、dbed_vmclonedb コマンドは次のような WARNING メッセージおよび INFO メッセージを表示します。

```
WARNING: Not all tempfiles were included in snapshot for
$ORACLE_SID, there is no snapshot volume for
/clone_path/temp02.dbf.
WARNING: Could not recreate tempfiles for $ORACLE_SID due to
lack of free space.INFO: The sql script for adding tempfiles to
$ORACLE_SID is at /tmp/add_tf.$ORACLE_SID.sql.
```

   $ORACLE_SID は、クローンデータベースの名前です。

2. Oracle 一時ファイルを再作成するために、add_tf.*$ORACLE_SID*.sql という名前のスクリプトが /tmp ディレクトリに格納されています。このスクリプトには、消失した一時ファイルを再作成できる SQL*Plus コマンドが含まれています。

3. /tmp/add_tf.*$ORACLE_SID*.sql スクリプトのコピーを作成して開き、消失した一時ファイルの一覧を表示します。

   次に、add_tf.*$ORACLE_SID*.sql スクリプトの例を示します。

```
$ cat add_tf.mnt2.sql
-- Commands to add tempfiles to temporary tablespaces.
-- Online tempfiles have complete space information.
-- Other tempfiles may require adjustment.
ALTER TABLESPACE TEMP ADD TEMPFILE
 'DBED: ADJUST_BEFORE_RUN/snap_data11r2/mnt2/temp01.dbf'
SIZE 20971520  REUSE AUTOEXTEND ON NEXT 655360  MAXSIZE 32767M;
```

4. 一時ファイルを再作成する必要性の有無を判断します。一時ファイルを再作成する場合は、次の手順に進みます。

5. add_tf.*$ORACLE_SID*.sql ファイルで、必要に応じて一時ファイルのサイズおよびデフォルトのパス名を編集し、データベースストレージで設定したクローンボリュームに保存します。

警告: このスクリプトは、必ず編集を行ってから実行してください。未編集の状態では、パス名が無かったり、指定したマウントポイントに十分な空き領域が割り当てられていないことがあるためです。

---

6 `add_tf.$ORACLE_SID.sql` スクリプトを変更した後で、クローンデータベースに対して実行します。

7 スクリプトが正常に実行されたら、そのスクリプトは削除してもかまいません。

スナップショットとデータベースの再同期

クローンデータベースをすでに使い終わったときや更新したいときは、クローンデータベースを元のデータベースと再同期化させることができます。これは、スナップショットボリュームの更新、または、分割したスナップショットイメージの現在のデータベースイメージへのマージとも呼ばれます。再同期した後でも、バックアップや意思決定支援を行うためにスナップショットを再度作成できます。

1 つのボリュームでデータを再同期化させる方法には次があります。

- 元のボリュームからのスナップショットの再同期化。この手順については、この項で説明します。
- スナップショットから元のボリュームを再同期します。この方法を逆再同期と呼びます。逆再同期は破損したデータベースのリストアに必要なことがあり、通常は、バックアップメディアからの完全なリストアなどの別の方法を使うよりも速く実行できます。
  p.165 の「ボリュームレベルのスナップショット（FlashSnap）の逆再同期について」を参照してください。
  単一インスタンス Oracle では、クローンデータベースは、スナップショットのデータから元のボリュームを逆再同期化させるのに使うこともできるほか、今後の使用に備えて、元のボリュームにスナップショットボリュームを再結合することにより（つまり、スナップショットボリュームを再同期化することにより）、破棄することもできます。

メモ: 逆再同期オプションは Oracle RAC 環境では使えません。

元のボリュームからスナップショットを再同期化できます。

前提条件

- Oracle データベース管理者としてログインする必要があります。
- スナップショットイメージの再同期前に、スナッププランの検証とスナップショットの作成が必要です。

  p.394 の「データベーススナップショットの作成について」を参照してください。

  p.414 の「スナッププランの検証（dbed_vmchecksnap）」を参照してください。

  p.419 の「スナップショットの作成（dbed_vmsnap）」を参照してください。
- クローンデータベースを作成した場合は、`dbed_vmclonedb -o umount` コマンドを使って、クローンデータベースのシャットダウンとファイルシステムのマウント解除を行います。プライマリホストとセカンダリホストが異なる場合は、このコマンドを使ってディスクグループをデポートすることもできます。

  p.430 の「クローンデータベースの停止とファイルシステムのマウント解除」を参照してください。
- Oracle データベースは、必須アーカイブの記録先を少なくとも 1 つ保持している必要があります。

使用に関する注意事項

- `dbed_vmsnap` コマンドは、どちらのホストでも実行できます。
- オフホスト構成では、`dbed_vmsnap` コマンドは、セカンダリホストからデポートされたディスクグループをインポートし、ディスクグループを元のディスクグループに結合します。スナップショットボリュームは、再び元のボリュームのプレックスになります。その後、スナップショットが再同期化されます。
- 詳しくは、`dbed_vmsnap`（1M）のマニュアルページを参照してください。

メモ: 次の手順では、Oracle データベース管理者としてコマンドを実行する必要があります。

スナップショットイメージを再同期化させるには

◆ 次のように dbed_vmsnap コマンドを使います。

```
$ /opt/VRTS/bin/dbed_vmsnap -S ORACLE_SID -f SNAPPLAN -o resync
```

この例では、スナップショットイメージがプライマリデータベースと再同期処理を行います。

```
$ dbed_vmsnap -S FLAS11r2 -f snap -o resync

SFDB vxsfadm WARNING V-81-0000 dbed_vmsnap will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0000 As an altername use /opt/VRTS/bin/vxsfadm
resync started at Mon Aug 19 07:30:42 2013.
Importing snapshot diskgroups ... Done
Joining snapshot diskgroups to original diskgroups ... Done
Reattaching snapshot volumes to original volumes ... Done
resync ended at Mon Aug 19 07:31:13 2013.
```

これで、再びスナップショットの作成を開始できるようになりました。

スナップショットボリュームの削除

スナップショットボリュームが不要となった場合は、そのスナップショットボリュームを削除し、vxedit rm コマンドを使って他の用途のためにディスク領域を開放します。

前提条件
- スーパーユーザーとしてログインする必要があります。
- ボリュームが、マウントされているファイルシステム上にある場合は、ファイルシステムをマウント解除してからボリュームを削除します。

スナッププランとスナップショットボリュームを削除するには

1 スナップショットを削除して使用済みのストレージを開放します。

- スナップショットが作成されていない場合、次のようにスナップショットを削除します。

```
# vxsnap -g diskgroup dis snapshot_volume
# vxvol -g diskgroup stop snapshot_volume
# vxedit -g diskgroup -rf rm snapshot_volume
```

- スナップショットを作成しておらず、スナップショットプレックス(ミラー)が存在する場合は、次のようにスナップショットを作成します。

```
# vxsnap -g diskgroup rmmir volume
```

2 DCO と DCO ボリュームを次のように削除します。

```
# vxsnap -g diskgroup unprepare volume
```

3 スナッププランを削除します。

```
# /opt/VRTS/bin/dbed_vmchecksnap -S PROD -f snapplan -o remove
```

たとえば、次のコマンドを使うと、スナップショットボリュームがディスクグループ PRODdg から削除されます。

```
# vxsnap -g PRODdg dis snap_v1
# vxvol -g PRODdg stop snap_v1
# vxedit -g PRODdg -rf rm snap_v1
```

Oracle リカバリのガイドライン

最適な Oracle リカバリを行うには、次の手順を実行する必要があります。

- 「Storage Rollback 前の全制御ファイルのバックアップ」
- 「制御ファイルがロールバックされないことの確認」
- 「すべてのアーカイブ REDO ログが利用可能であることの確認」
- 「メディアのリカバリ手順」

Storage Rollback 前の全制御ファイルのバックアップ

その後の Oracle リカバリが正常に行われない場合は、このガイドラインを使ってください。

Oracle 社では、Oracle データベースごとに制御ファイルのコピーを少なくとも 2 部作成し、それぞれ異なるディスクに保存することを推奨しています。制御ファイルは、データベース構造を変更する前後にもバックアップを作成する必要があります。

メモ: `dbed_ckptcreate` コマンドを実行すると、Storage Checkpoint の作成時に制御ファイルとログ情報が自動的に保存されます。

制御ファイルがロールバックされないことの確認

制御ファイルは、データベースの構造を記述する小さいバイナリファイルで、データベースをマウント、オープンおよび保持する際に必要です。制御ファイルに格納されるのは、

必要なすべてのデータベースファイル情報、ログファイル情報、データベース名、データベース作成のタイムスタンプおよび修復に必要な Storage Checkpoint やログシーケンスの情報などの同期化情報です。

制御ファイルをロールバックすると、データベースの物理構造と制御ファイルの間に不整合が起きる場合があります。

すべてのアーカイブ REDO ログが利用可能であることの確認

データベースの完全なリカバリを行うには、オンライン状態のデータベースバックアップとアーカイブログが必要です。

すべてのアーカイブログ情報を一覧表示するには V$ARCHIVED_LOG で、アーカイブ先の場所を一覧表示するには V$ARCHIVE_DEST でクエリーを実行します。

メモ: アーカイブされた情報の問い合わせについて詳しくは、Oracle のマニュアルを参照してください。

Storage Foundation for Oracle RAC の場合、アーカイブログの記録先は Veritas クラスタファイルシステム上に存在する必要があります。

必要なアーカイブ REDO ログファイルをリストアするために、V$LOG_HISTORY に対するクエリーを行い、すべてのアーカイブ REDO ログ履歴を表示することも、V$RECOVERY_LOG に対するクエリーを行い、リカバリに必要なアーカイブ REDO ログのみを表示すせることもできます。必要なアーカイブ REDO ログファイルは、LOG_ARCHIVE_DEST パラメータに指定された宛先または別の位置にリストアできます。アーカイブ REDO ログが別の場所にリストアされた場合は、メディアのリカバリ時に `ALTER DATABASE RECOVER ... FROM` 文を使います。

Storage Rollback の実行後、Oracle リカバリを実行し、アーカイブ REDO ログの一部または全部を適用します。

メモ: データベース(制御ファイルと REDO ログを含む)を Storage Checkpoint にロールバックした後で、Oracle データベースインスタンスをリカバリする必要があります。データベースのロールフォワードはサポートされていません。つまり、アーカイブされた REDO ログを適用することはできません。

メディアのリカバリ手順

次に、メディアの完全なリカバリと部分的なリカバリの手順を示します。

メディアのリカバリ手順

- メディアの完全なリカバリを実行するには、次のように入力します。

```
SQL> SET AUTORECOVERY ON;

SQL> RECOVER DATABASE;
```

- 部分的なメディアリカバリを実行するには、次のいずれかを使います。

```
SQL> RECOVER DATABASE UNTIL CANCEL;
```

または

```
SQL> RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss';
```

（../bdump/alert*.log ファイルを確認することでエラーの時間を確認できます。）
または

```
SQL> RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss'¥
using backup controlfile;
```

または

```
SQL> RECOVER DATABASE UNTIL CHANGE scn;
```

- 不完全メディアリカバリの実行後にデータベースを開くには、次のものを使います。

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS はログシーケンスをリセットします。不完全メディアリカバリの実行後は、RESETLOGS オプションが必要です。RESETLOGS オプションを使ってデータベースを開いた後、ロールバックしたばかりの Storage Checkpoint と、それより前に取得された Storage Checkpoint をすべて削除します。これらの以前の Storage Checkpoint は Storage Rollback に使えなくなっています。これらの Storage Checkpoint を削除した後に、新しい Storage Checkpoint を作成します。

警告: 同じ Storage Checkpoint に複数回ロールバックを試みると、データが破損する可能性があります。ロールバック後は、ロールバックした Storage Checkpoint を削除し、新しい Storage Checkpoint を作成します。

リカバリの詳細については、Oracle のマニュアルを参照してください。

Database Storage Checkpoint のコマンド

Storage Foundation for Databases（SFDB）ツールでは、コマンドラインインターフェースから多数の主要な操作を行えます。コマンドラインインターフェースによって、コマンド操作をスクリプトやその他の管理処理に組み入れることができます。

サポートされる SFDB ツールのコマンドは /opt/VRTS/bin ディレクトリにあります。

このコマンドのオンラインマニュアルページは、/opt/VRTS/man ディレクトリに格納されています。

メモ: Storage Foundation 製品のデータベースでは、このコマンドを実行するときは、`$ORACLE_SID` を `$ORACLE_SID=instance_name` に置き換え、インスタンスが実行中のインスタンス名を指定してください。

表 32-3 に、コマンドラインで利用可能なコマンドについてまとめます。

表 32-3 Database Checkpoint のコマンド

コマンド	説明
`dbed_ckptcreate`	Oracle データベースの Storage Checkpoint を作成するコマンド。 p.441 の「dbed_ckptcreate を使った Storage Checkpoint の作成」を参照してください。
`dbed_ckptdisplay`	Oracle インスタンスに関連付けられている Storage Checkpoint を表示するコマンド。 p.443 の「dbed_ckptdisplay を使った Storage Checkpoint の表示」を参照してください。
`dbed_ckptmount`	Oracle インスタンスの Storage Checkpoint をマウントするコマンド。 p.444 の「dbed_ckptmount を使った Storage Checkpoint のマウント」を参照してください。
`dbed_ckptumount`	Oracle インスタンスの Storage Checkpoint をマウント解除するコマンド。 p.446 の「dbed_ckptumount を使った Storage Checkpoint のマウント解除」を参照してください。
`dbed_ckptrollback`	Oracle インスタンスを Storage Checkpoint の特定時点におけるイメージにロールバックするコマンド。 p.446 の「dbed_ckptrollback を使った Storage Rollback の実行」を参照してください。

コマンド	説明
dbed_ckptremove	Oracle インスタンスの Storage Checkpoint を削除するコマンド。 p.448 の「dbed_ckptremove を使った Storage Checkpoint の削除」を参照してください。
dbed_clonedb	既存すべてのデータベースファイルのクローン作成で Oracle データベースのコピーを作成し、制御ファイルを適切に再作成するコマンド。クローン作成されたこのデータベースは、既存のデータベースと同じホストで別の SID を使用する場合にかぎり起動できます。 p.448 の「dbed_clonedb を使った Oracle インスタンスのクローン作成」を参照してください。

この項では、Storage Checkpoint と Storage Rollback の管理作業を行うための SFDB コマンド例を示します。それぞれのコマンドを使う前に、コマンドごとに示される前提条件と使用に関する注意事項を確認してください。

メモ: 各コマンドとその構文、および利用可能なオプションについて詳しくは、個々のマニュアルページを参照してください。

dbed_update を使ったリポジトリの作成または更新

dbed_update コマンドを使って、リポジトリを作成または更新できます。

メモ: データファイルの追加、削除、PFILE から SPFILE への変換、SPFILE から PFILE への変換など、データベースの構造を変更するたびに、dbed_update コマンドを実行する必要があります。

リポジトリの作成前または更新前に、次の条件を満たす必要があります。

前提条件

- データベース管理者としてログインする必要があります(通常、ユーザー ID は oracle)。

使用に関する注意事項

- リポジトリがすでに存在する場合は、dbed_update コマンドによってこの情報が更新されます。
- データベースは事前に起動しておく必要があり、可変の引数 *ORACLE_SID* と *ORACLE_HOME* には -S オプションと -H オプションをそれぞれ指定する必要があります。
- 詳しくは dbed_update(1M)のマニュアルページを参照してください。

リポジトリを更新するには

◆ 次のように、dbed_update コマンドを使います。

```
$ /opt/VRTS/bin/dbed_update -S PROD -H /oracle/product/ORA_HOME
```

メモ: ユーザーが SPFILE から PFILE へ切り替える場合、-P オプションを使用して `dbed_update` コマンドを発行する必要があります。<ORACLE_PFILE > オプションは Oracle PFILE が ORACLE_HOME/dbs ディレクトリにない場合、Oracle PFILE の完全修飾パスを指定します。

---

dbed_ckptcreate を使った Storage Checkpoint の作成

`dbed_ckptcreate` コマンドを使って、コマンドラインから Oracle データベースの Storage Checkpoint を作成できます。

Storage Checkpoint は online、offline、instant のいずれかです。デフォルトでは、Storage Checkpoint は online に設定されています。online の Storage Checkpoint の場合、Storage Checkpoint の作成時にデータベースがホットバックアップモードになります。offline を指定した場合、データベースは停止していると見なされます。

instant を指定する場合、データベースはオンラインである必要があり、Storage Checkpoint は「クラッシュ回復」型の回復として取得されます。

Storage Checkpoint を作成するには、次の条件を満たす必要があります。

前提条件

- データベース管理者としてログインする必要があります（通常、ユーザー ID は oracle）。
- リカバリ機能を最大限に活用するには、Storage Checkpoint の作成時に、ARCHIVELOG モードを常に有効にしておきます。

使用に関する注意事項

- dbed_ckptcreate は Storage Checkpoint の情報を SFDB リポジトリに格納します。
- 詳しくは `dbed_ckptcreate`（1M）のマニュアルページを参照してください。

データベースがオンライン中に Storage Checkpoint を作成するには

◆ 次のように、dbed_ckptcreate コマンドを使います。

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD ¥
-H /oracle/product/ORA_HOME -o online

SFDB vxsfadm WARNING V-81-0962 dbed_ckptcreate
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an altername
use /opt/VRTS/bin/vxsfadm
Putting database in backup mode...                          Done
Creating Storage Checkpoint Ckpt_1378514849 ...             Done
Storage Checkpoint Ckpt_1378514849 created
Removing the database from backup mode...                   Done
```

データベースがオフライン中に Storage Checkpoint を作成するには

◆ 次のように、dbed_ckptcreate コマンドを使います。

```
$/opt/VRTS/bin/dbed_ckptcreate -S PROD ¥
-H /oracle/product/ORA_HOME -o offline

SFDB vxsfadm WARNING V-81-0962 dbed_ckptcreate will be
 deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
 /opt/VRTS/bin/vxsfadm
Creating Storage Checkpoint Ckpt_1378465225 ...             Done
Storage Checkpoint Ckpt_1378465225 created
```

インスタント Storage Checkpoint を作成するには

◆ データベースがオンラインになっていることを確認し、dbed_ckptcreate コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptcreate -S PROD ¥
-H /oracle/product/oracle -o instant

SFDB vxsfadm WARNING V-81-0962 dbed_ckptcreate
will be deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
 /opt/VRTS/bin/vxsfadm
Creating Storage Checkpoint Ckpt_1378465458 ...             Done
Storage Checkpoint Ckpt_1378465458 created
```

dbed_ckptdisplay を使った Storage Checkpoint の表示

dbed_ckptdisplay コマンドを使って、コマンドラインから Oracle データベースに関連付けられた Storage Checkpoint を表示できます。

Storage Checkpoint を表示するには、次の条件を満たす必要があります。

前提条件

- データベース管理者としてログインする必要があります。

使用に関する注意事項

- dbed_ckptdisplay は、SF Enterprise 製品 で作成された Storage Checkpoint を表示するだけでなく、その他の Storage Checkpoint（たとえば、Capacity Planning ユーティリティと NetBackup によって作成された Storage Checkpoint）も表示します。
- 状態フィールドにより、Storage Checkpoint が部分（P）、完了（R）、エラー（E）、マウント済み（M）、読み取り専用（R）、書き込み可能（W）のどの状態にあるのか、また、Storage Checkpoint のタイプがオンライン（ON）、オフライン（OF）、インスタント（IN）、不明（UN）のどのタイプであるのかを確認できます。
- Database FlashSnap コマンドは Storage Checkpoint 機能に統合されています。スナップショットボリュームから引き継いだ Storage Checkpoint をセカンダリホストに表示およびマウントできます。
- 詳しくは dbed_ckptdisplay（1M）のマニュアルページを参照してください。

Database Checkpoint を表示するには

◆ 次のように dbed_ckptdisplay コマンドを使って、SF Enterprise 製品 で作成された Storage Checkpoint の情報を表示します。

```
 $ /opt/VRTS/bin/dbed_ckptdisplay -S PROD ¥
-H /oracle/product/ORA_HOME


SFDB vxsfadm WARNING V-81-0962 dbed_ckptdisplay will be deprecated

 in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
/opt/VRTS/bin/vxsfadm
Storage Checkpoint       Creation time              Status
==============================================================
Ckpt_1378463767          Fri Sep  6 14:58:16 2013  C+R+ON
Ckpt_1378465225          Fri Sep  6 15:00:32 2013  C+R+OF
Ckpt_1378465458          Fri Sep  6 14:58:29 2013  C+R+IN
```

その他の Storage Checkpoint を表示するには

◆ dbed_ckptdisplay コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD ¥
-H /oracle/product/ORA_HOME -o other

SFDB vxsfadm WARNING V-81-0962 dbed_ckptdisplay will be deprecated

 in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
 /opt/VRTS/bin/vxsfadm
Storage Checkpoint                   Creator  Location
-----------------------------------  -------  --------
NetBackup_incr_PROD_95518740         NBU      /data
NetBackup_incr_PROD_95518673         NBU      /data
```

すべての Storage Checkpoint を表示するには

◆ dbed_ckptdisplay コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptdisplay -S PROD ¥
-H /oracle/product/ORA_HOME -o all


SFDB vxsfadm WARNING V-81-0962 dbed_ckptdisplay will be
 deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
 /opt/VRTS/bin/vxsfadm
Storage Checkpoint       Creation time             Status
==========================================================
Ckpt_1378463767          Fri Sep  6 14:58:16 2013  C+R+ON
Ckpt_1378465225          Fri Sep  6 15:00:32 2013  C+R+OF
Ckpt_1378465458          Fri Sep  6 14:58:29 2013  C+R+IN

Storage Checkpoint                   Creator  Location
-----------------------------------  -------  --------
NetBackup_incr_PROD_95518740         NBU      /data
NetBackup_incr_PROD_95518673         NBU      /data
```

dbed_ckptmount を使った Storage Checkpoint のマウント

dbed_ckptmount コマンドを使って、コマンドラインからデータベースの Storage Checkpoint をマウントできます。

Storage Checkpoint をマウントするには、次の条件を満たす必要があります。

前提条件

- データベース管理者としてログインする必要があります。

使用に関する注意事項

- dbed_ckptmount コマンドは、Storage Checkpoint をファイルシステムの名前空間にマウントする場合に使います。マウントされた Storage Checkpoint は、コンピュータ上ではファイルシステムとして表示されるため、通常のファイルシステムベースのすべてのコマンドで操作できます。
- Storage Checkpoint は読み取り専用、または読み取りおよび書き込み可能としてマウントできます。デフォルトでは、Storage Checkpoint は読み取り専用としてマウントされます。
- rw（読み取りおよび書き込み可能）オプションを指定した場合は、rw_xxxx（xxx は整数）が Storage Checkpoint 名に付け加えられます。
- 指定したマウントポイントディレクトリが存在しない場合、Oracle データベース所有者にこのディレクトリを作成する権限がある場合にかぎり、dbed_ckptmount コマンドは必要なディレクトリを作成してから Storage Checkpoint をマウントします。
- Database FlashSnap コマンドは Storage Checkpoint 機能に統合されています。スナップショットボリュームから引き継いだ Storage Checkpoint をセカンダリホストに表示およびマウントできます。
- 詳しくは dbed_ckptmount（1M）のマニュアルページを参照してください。

読み取りおよび書き込みオプションを指定して Storage Checkpoint をマウントするには

◆ dbed_ckptmount コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Ckpt_971672042 ¥
-m /tmp/ckpt_rw -o rw
SFDB vxsfadm WARNING V-81-0962 dbed_ckptmount will be
deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
/opt/VRTS/bin/vxsfadm
Creating Storage Checkpoint Ckpt_1378463767_rw_1378469718 ...
Done
Storage Checkpoint Ckpt_1378463767_rw_1378469718 created
Mounting Checkpoint to /tmp/ckptrw...
Done
```

読み取り専用オプションを指定して Storage Checkpoint をマウントするには

◆ dbed_ckptmount コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptmount -S PROD -c Ckpt_971672042 ¥
-m /tmp/ckpt_ro -o ro
```

dbed_ckptumount を使った Storage Checkpoint のマウント解除

dbed_ckptumount コマンドを使って、コマンドラインから Oracle データベースの Storage Checkpoint のマウントを解除できます。

Storage Checkpoint のマウントを解除するには、次の条件を満たす必要があります。

前提条件
- データベース管理者としてログインする必要があります。

使用に関する注意事項
- dbed_ckptumount コマンドは、ファイルシステムの名前空間から Storage Checkpoint のマウントを解除する場合に使います。マウントされた Storage Checkpoint は、コンピュータ上ではファイルシステムとして表示されるため、通常のファイルシステムベースのすべてのコマンドで操作できます。マウントされた Storage Checkpoint が必要なくなった場合、マウントを解除できます。
- 詳しくは dbed_ckptumount（1M）のマニュアルページを参照してください。

Storage Checkpoint のマウントを解除するには

◆ dbed_ckptumount コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptumount -S PROD -c Ckpt_97167042
SFDB vxsfadm WARNING V-81-0962 dbed_ckptumount will be
deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
/opt/VRTS/bin/vxsfadm
Unmounting the checkpoint...
Done
```

dbed_ckptrollback を使った Storage Rollback の実行

dbed_ckptrollback コマンドを使って、Oracle データベースを Storage Checkpoint にロールバックできます。

Storage Rollback を実行するには、次の条件を満たす必要があります。

前提条件
- データベース管理者としてログインする必要があります。

使用に関する注意事項

- dbed_ckptrollback コマンドは、Oracle データベースを指定の Storage Checkpoint にロールバックします。Storage Rollback は、データベース全体、特定の表領域、またはデータファイルのリストに対して実行できます。
  データベース全体のデータベースロールバックでは、Storage Rollback を開始する前にデータベースを停止する必要があります。
  dbed_ckptrollback コマンドは、Oracle データベースがアクティブの場合は開始しません。ただし、表領域またはデータファイルの Storage Rollback を実行するには、ロールバック対象の表領域またはデータファイルのみをオフラインにするだけでかまいません（データベース全体をオフラインにする必要はありません）。
- 詳しくは、dbed_ckptrollback（1M）のマニュアルページを参照してください。

Oracle データベースを Storage Checkpoint にロールバックするには

◆ dbed_ckptrollback コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD ¥
-H /oracle/product/ORA_HOME -c Ckpt_903937870
```

表領域を Storage Checkpoint を使ってロールバックするには

◆ 次のように、-T オプションを指定して dbed_ckptrollback コマンドを使います。

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD ¥
-H /oracle/product/ORA_HOME -T DATA01 -c Ckpt_903937870
```

Oracle データベースが稼動中の場合は、このコマンドを実行する前に、表領域をオフラインにする必要があります。表領域がオンラインの場合、このコマンドは失敗します。

メモ: TEMP 表領域はロールバックできません。

データファイルを Storage Checkpoint を使ってロールバックするには

◆ 次のように、-F オプションを指定して dbed_ckptrollback コマンドを使います。

```
$ /opt/VRTS/bin/dbed_ckptrollback -S PROD ¥
-H /oracle/product/ORA_HOME ¥
-F /share/oradata1/data01.dbf,/share/oradata2/index01.dbf ¥
-c Checkpoint_903937870
```

Oracle データベースが稼動中の場合は、このコマンドを実行する前に、データファイルをオフラインにする必要があります。データファイルがオンラインの場合、このコマンドは失敗します。

dbed_ckptremove を使った Storage Checkpoint の削除

dbed_ckptremove コマンドを使って、コマンドラインから Oracle データベースの Storage Checkpoint を削除できます。

Storage Checkpoint を削除するには、次の条件を満たす必要があります。

前提条件
- データベース管理者としてログインする必要があります。

使用に関する注意事項
- dbed_ckptremove コマンドは、1 つのファイルシステム、または関連付けられた複数のファイルシステムから Storage Checkpoint を削除する場合に使います。Storage Checkpoint は、dbed_ckptcreate（1M）コマンドを使って作成しておく必要があります。
- 詳しくは dbed_ckptremove（1M）のマニュアルページを参照してください。

Storage Checkpoint を削除するには

◆ dbed_ckptremove コマンドを次のように使います。

```
$ /opt/VRTS/bin/dbed_ckptremove -S PROD ¥
-c Ckpt_971672042
```

dbed_clonedb を使った Oracle インスタンスのクローン作成

dbed_clonedb コマンドを使うと、Storage Checkpoint から Oracle インスタンスのクローンを作成できます。

Storage Checkpoint を使って既存のデータベースのクローンを作成する場合、同一ホスト上に作成する必要があります。

警告: AIX の場合:

mount コマンドで -o cio オプションを使ってプライマリデータベースファイルシステムをマウントする場合は、dbed_clonedb を使ってクローンデータベースを作成するときに同時 I/O は保存されません。

Oracle インスタンスのクローンを作成する前に、次の条件を満たす必要があります。

前提条件

- まず、Storage Checkpoint を作成する必要があります。p.441 の「dbed_ckptcreate を使った Storage Checkpoint の作成」を参照してください。
- データベース管理者としてログインする必要があります。
- クローンデータベースの作成に必要な十分な領域とシステムリソースがシステムにあることを確認してください。
- クローンデータベースは、プライマリデータベースと同量のメモリとマシンリソースを消費します。

使用に関する注意事項

- dbed_clonedb コマンドは、データベースのコピーを作成し、すべての既存データベースファイルのクローンを新しい場所に作成する場合に使います。
- *ORACLE_SID* と *ORACLE_HOME* 環境変数がプライマリデータベースに設定されている必要があります。
- 詳しくは、dbed_clonedb(1M)のマニュアルページを参照してください。

Oracle RAC の制限事項

- Storage Checkpoint を使ってデータベースをクローン作成する場合は、ノードには同じ Oracle RAC のクラスタにあるすべてのノードを指定できますが、CFS ファイルシステム上にアーカイブログの記録先が必要です。そうしない場合は、アーカイブログファイルを手動でコピーする必要があります。

「表 32-4」に dbed_clonedb コマンドのオプションの一覧を表示します。

表 32-4 dbed_clonedb コマンドオプション

オプション	説明
-S *CLONE_SID*	新しい Oracle SID の名前を指定します。この名前が、新しいデータベースインスタンスの名前になります。
-m *MOUNT_POINT*	Storage Checkpoint の新しいマウントポイントを指定します。
-c *CKPT_NAME*	Storage Checkpoint の名前を指定します。
-o umount	クローンデータベースを停止し、Storage Checkpoint ファイルシステムのマウントを解除します。

オプション	説明
-o restartdb	Storage Checkpoint ファイルシステムをマウントし、クローンデータベースを起動します。-o restartdb オプションを指定すると、クローンデータベースのリカバリは試行されません。
-d	-o umount オプションと組み合わせて使います。-d オプションを指定した場合、クローンデータベースの作成に使われた Storage Checkpoint がクローンデータベースと一緒に削除されます。
-p	修正する、または起動前にクローンデータベースの初期化パラメータファイルに追加する初期化パラメータが含まれるファイルを指定します。pfile_modification_file の書式は、Oracle の初期化パラメータファイルの書式と同じです。

Oracle インスタンスのクローンを作成するには

◆ 次のように dbed_clonedb コマンドを使います。

```
$/opt/VRTS/bin/dbed_clonedb -S new2 -m /tmp/new2 ¥
-c onCkpt
SFDB vxsfadm WARNING V-81-0962 dbed_clonedb will
 be deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative
use /opt/VRTS/bin/vxsfadm
Creating Storage Checkpoint onCkpt_rw_1378467816 ... Done
Storage Checkpoint onCkpt_rw_1378467816 created
Mounting Checkpoint to /tmp/new2...
Done
Cloning the Pfile ...
Done
Mounting the database...
Done
Recovering the clone database...
```

クローンデータベースを停止し、Storage Checkpoint のマウントを解除するには

◆ 次のように dbed_clonedb コマンドを使います。

```
$ /opt/VRTS/bin/dbed_clonedb -S new2 -o umount
SFDB vxsfadm WARNING V-81-0962 dbed_clonedb will
be deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative
use /opt/VRTS/bin/vxsfadm
Shutting down the clone database new2... Done
Unmounting the checkpoint... Done
```

Storage Checkpoint ファイルシステムをマウントし、クローンデータベースを起動するには

◆ 次のように dbed_clonedb コマンドを使います。

```
$/opt/VRTS/bin/dbed_clonedb -S new2 -o restartdb
SFDB vxsfadm WARNING V-81-0962 dbed_clonedb will be
 deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
 /opt/VRTS/bin/vxsfadm
Mounting Checkpoint to /tmp/new2... Done
Mounting the database... Done
Starting the Clone database ... Done
```

クローンデータベースとクローンデータベースの作成に使った **Storage Checkpoint** を削除するには

◆ 次のように、dbed_clonedb コマンドを使います。

```
$ /opt/VRTS/bin/dbed_clonedb -S new2 -o umount -d
SFDB vxsfadm WARNING V-81-0962 dbed_clonedb will be
 deprecated in the next release
SFDB vxsfadm WARNING V-81-0963 As an alternative use
 /opt/VRTS/bin/vxsfadm
Shutting down the clone database new2... Done
Unmounting the checkpoint... Done
Deleting the Checkpoint onCkpt_rw_1316003740...
Deleting Checkpoint from /db01... Done
```

9

参照先

- 付録 A. VCS Oracle エージェント
- 付録 B. クラスタ化された配備のサンプル設定ファイル
- 付録 C. Database FlashSnap の状態情報
- 付録 D. サードパーティソフトウェアによるファイルのバックアップ

A

VCS Oracle エージェント

この付録では以下の項目について説明しています。

- Oracle 用の VCS エージェント

Oracle 用の VCS エージェント

Oracle 用の VCS エージェントには、Oracle の高可用性を実現するために連携して働く次のエージェントが含まれています。

- Oracle エージェントは データベースプロセスを監視します。
  p.453 の「Oracle エージェント関数」を参照してください。
  p.457 の「Oracle エージェントのリソースタイプ定義」を参照してください。
- Netlsnr エージェントはリスナープロセスを監視します。
  p.462 の「Netlsnr エージェント関数」を参照してください。
- ASMDG エージェントは Oracle ASM ディスクグループを監視します。

エージェント機能とリソースタイプについて詳しくは、『Cluster Server Agent for Oracle インストールおよび設定ガイド』を参照してください。

Oracle エージェント関数

Oracle エージェントはデータベースプロセスを監視します。

表 A-1 に、Oracle エージェント関数を一覧で示します。

表 A-1　Oracle エージェント関数

エージェントの操作	説明
Monitor	Oracle プロセスの状態を確認します。Oracle エージェントでは、基本と詳細の 2 段階の監視レベルを設定できます。 p.454 の「従来のデータベースおよびコンテナデータベース内の Oracle エージェントに対する監視オプション」を参照してください。
Info	データベースの状態について静的または動的な情報を表示します。 **メモ:** この属性は PDB リソースには適用されません。 p.455 の「Cluster Server agent for Oracle の Info エントリポイント」を参照してください。
Action	リソースで事前定義済みの処理を実行します。 p.456 の「Cluster Server agent for Oracle の Action エントリポイント」を参照してください。

従来のデータベースおよびコンテナデータベース内の Oracle エージェントに対する監視オプション

Oracle エージェントでは、基本と詳細の 2 段階の監視レベルを設定できます。デフォルトでは、エージェントは基本監視を実行します。

基本監視モードでは、次のオプションが用意されています。

- プロセスチェック
- 診断

Oracle リソースの MonitorOption 属性は、エージェントがプロセスチェックモードまたは診断モードで、基本監視を実行する必要があるかどうかを決定します。

表 A-2には、基本監視オプションについて説明されています。

表 A-2　基本監視オプション

オプション	説明
0 (デフォルト)	プロセスチェック エージェントでは、`ora_dbw0`、`ora_smon`、`ora_pmon`、`ora_lmon`、`ora_lgwr` の各プロセスについてのプロセステーブルをスキャンして、Oracle が稼働していることを確認します。

オプション	説明
1	診断 エージェントは Oracle からの診断用 API を使って、SGA を監視し、インスタンスに関する情報を取得します。

基本監視を設定する場合は、次の注意事項を確認します。

- Oracle プロセスの基本監視はユーザーに固有です。したがって、別のユーザー環境で起動された Oracle インスタンスをオンラインとして検出することはできません。たとえば、ある Oracle インスタンスがユーザー oraVRT の環境で起動されており、エージェントがユーザー oracle 用に設定されていると、そのエージェントは oraVRT の環境で起動されたインスタンスをオンラインとして検出しません。
  これにより、ノード 上のリソースをオンラインにするために発行したコマンドによって、その ノード (または別の ノード) 上ですでに実行中のインスタンスがオンライン化されるという状況になる場合があります。
  そのため、Veritas InfoScale 製品 の制御外で起動するインスタンスには、そのインスタンスの OS ユーザーに対応した正しい Owner 属性を設定することを推奨します。

詳細監視モードでは、エージェントはデータベースのテストテーブルでトランザクションを実行し、Oracle データベースが正常に機能していることを確認します。エージェントはこのテストテーブルを内部の目的で使用します。テストテーブルではその他のトランザクションを実行しないことをお勧めします。

Cluster Server agent for Oracle の Info エントリポイント

Info エントリポイントをサポートする で、データベースの状態に関する静的または動的な情報を表示します。

Info エントリポイントを呼び出すには、次のコマンドを入力します。

```
# hares -value resource ResourceInfo [system]¥
 [-clus cluster | -localclus]
```

上記のコマンドを正常に実行するには、次のコマンドを実行する必要があります:

- 次のコマンドを使って、リソースの refreshinfo スクリプトを実行します。
  ```
  # hares -refreshinfo <res-name> -sys <sys-name>
  ```

このエントリポイントでは、次の静的な情報を取得します。

- バージョン
- インスタンス番号（InstanceNo）
- インスタンス名（InstanceName）
- データベース名（DatabaseName）
- ホスト名（HostName）
- 開始時間（StartupTime）

- パラレル（Parallel）
- スレッド（Thread）
- インスタンスの役割（InstanceRole）

このエントリポイントでは、次の動的な情報を取得します。

- インスタンスの状態（InstanceStatus）
- ログイン情報（Logins）
- オープンモード（OpenMode）
- ログモード（LogMode）
- シャットダウンの待機（ShutdownPending）
- データベースの状態（DatabaseStatus）
- 共有プールの空き割合（Shared Pool Percent free）
- バッファ使用率（Buffer Hits Percent）

SQL 文を /opt/VRTSagents/ha/bin/Oracle/resinfo.sql に追加して、属性を加えることができます。次に例を示します。

```
select 'static:HostName:'||host_name from v$instance;
select 'dynamic:ShutdownPending:'||shutdown_pending from
v$instance;
```

選択したレコードの型は、次のようになります。

```
attribute_type:userkey_name:userkey_value
```

変数 *attribute_type* は、静的または動的のいずれかか、両方の性質を持った値にできます。

Cluster Server agent for Oracle の Action エントリポイント

は Action エントリポイントをサポートし、リソース上で事前定義済みの処理を実行できるようにします。

リソースで処理を実行するには、次のコマンドを入力します。

エージェントのカスタム処理を追加することもできます。

表 A-3 に、エージェントの事前定義済み処理の説明を示します。

表 A-3　事前定義済みのエージェント処理

アクション	説明
VRTS_GetInstanceName	すでに設定済みのインスタンスからそのインスタンス名を取得します。このオプションは、Oracle リソースと NetIsnr リソースに使うことができます。

アクション	説明
VRTS_GetRunningServices	エージェントが監視する処理の一覧を取得します。このオプションは、Oracle リソースと Netlsnr リソースに使うことができます。
DBRestrict	データベースのセッションを変更して、RESTRICTED モードを有効にします。
DBUndoRestrict	データベースのセッションを変更して、RESTRICTED モードを無効にします。
DBSuspend	データベースを一時停止します。
DBResume	一時停止していたデータベースを再開させます。
DBTbspBackup	表領域のバックアップを実行します。バックアップを作成する表領域名は、`actionargs` に含まれています。

メモ: VRTS_GetRunningServices、DBRestrict、DBResume、DBSuspend、DBTbspBackup および DBUndoRestrict の action エントリポイントは Oracle PDB リソースではサポートされません。

Oracle エージェントのリソースタイプ定義

の Oracle エージェントは、Veritas InfoScale 製品 では Oracle リソースタイプとして表されます。

次の OracleTypes.cf ファイルの抜粋は、Oracle リソースのタイプ定義を示します。

Oracle エージェントの属性の定義

Oracle エージェントの属性は、必須、オプション、内部に分類されます。

次の必須属性には、値を割り当てる必要があります。

表 A-4 Oracle エージェントの必須属性

必須属性	データ形式と値のタイプ	定義
SID	文字列 - スカラー	Oracle インスタンスを表す $ORACLE_SID 変数。Oracle エージェントと Oracle データベースサーバーでは Sid の大文字小文字が区別されます。 ポリシー管理データベースでは、Sid の属性は Sid プレフィックスに設定する必要があります。

必須属性	データ形式と値のタイプ	定義
Owner	文字列 - スカラー	データベースインスタンスを開始または停止する権限がある Oracle ユーザー。 エージェントでは、Oracle ユーザーとして LDAP ユーザーもサポートされます。
Home	文字列 - スカラー	Oracle バイナリファイルと設定ファイルへの $ORACLE_HOME パス。たとえば、パスとして /opt/ora_home を指定できます。 **メモ:** パスの最後にスラッシュ (/) は付けないでください。

次のオプションの属性は、設定に必要な場合に設定できます。

表 A-5 Oracle エージェントのオプションの属性

オプション属性	データ形式と値のタイプ	定義
StartUpOpt	文字列 - スカラー	Oracle インスタンスの起動オプション。この属性は従来のデータベースおよびコンテナデータベースに対しては次の値をとります。 ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ RECOVERDB ■ SRVCTLSTART ■ CUSTOM ■ SRVCTLSTART_FORCE デフォルトは STARTUP_FORCE です。 プラグ可能データベース内の Oracle インスタンスの起動オプション。この属性は次の値をとることができます。 ■ STARTUP ■ STARTUP_FORCE ■ RESTRICTED ■ OPEN_RO

オプション属性	データ形式と値のタイプ	定義
ShutDownOpt	文字列 - スカラー	Oracle インスタンスの停止オプション。この属性は従来のデータベースおよびコンテナデータベースに対しては次の値をとります。 ■ IMMEDIATE ■ TRANSACTIONAL ■ SRVCTLSTOP ■ CUSTOM ■ SRVCTLSTOP_TRANSACT ■ SRVCTLSTOP_ABORT ■ SRVCTLSTOP_IMMEDIATE デフォルトは IMMEDIATE です。 プラグ可能データベース内の Oracle インスタンスのシャットダウンオプションは IMMEDIATE です。
EnvFile	文字列 - スカラー	エントリポイントスクリプトによって提供されるファイルの絶対パス名。このファイルには、Oracle データベースサーバー環境のユーザーが設定する LD_LIBRARY_PATH、NLS_DATE_FORMAT などの環境変数が含まれます。 ファイル内容の構文は、Owner のログインシェルによって異なります。ファイルは Owner が読み取ることができなければなりません。ファイルにユーザー入力を求めるプロンプトが含まれていてはいけません。
Pfile	文字列 - スカラー	起動プロファイルの絶対パスを指定した初期化パラメータの名前。 また、サーバーパラメータファイルを使用することもできます。SPFILE パラメータのみを含むオンラインのテキスト初期化パラメータファイルを作成します。詳しくは、Oracle のマニュアルを参照してください。 **メモ:** この属性は従来のデータベースおよびコンテナデータベースにのみ適用されます。
AutoEndBkup	ブール - スカラー	AutoEndBkup 属性をゼロ以外の値に設定すると、オンライン時にデータベースのデータファイルをバックアップモードから解除します。 デフォルト: 1 **メモ:** コンテナデータベースまたはプラグ可能データベースのホットバックアップ中にノードに障害が発生した場合は、対応する CDB リソースの AutoEndBkup 属性を 1 に設定する必要があります。CDB の AutoEndBkup が 1 に設定されている場合、オンライン時に CDB と PDB の両方のバックアップも終了します。

オプション属性	データ形式と値のタイプ	定義
MonitorOption	整数 - スカラー	Oracle インスタンスの監視オプション。この属性には 0 または 1 を指定できます。 従来のデータベースおよびコンテナデータベースの場合: ■ 0 - プロセスチェック監視 (推奨) ■ 1 - 診断監視 プラグ可能データベースの場合: ■ 0 - 基本監視 デフォルト: 0 この属性値を 0 に設定した場合にのみ、エージェントは従来の DB および CDB に対するインテリジェントなリソース監視をサポートします。 p.454 の「従来のデータベースおよびコンテナデータベース内の Oracle エージェントに対する監視オプション」を参照してください。
MonScript	文字列 - スカラー	詳細な監視のために提供されているスクリプトへのパス名。デフォルト (基本監視) では、データベース PID のみが監視されます。 **メモ:** MonScript 属性の値が無効であったり、空の文字列を設定している場合、詳細監視は無効になります。 提供されている詳細監視スクリプトへのパス名は /opt/VRTSagents/ha/bin/Oracle/SqlTest.pl です。 また、MonScript には /opt/VRTSagents/ha への相対パス名も指定できます。相対パス名は、パス ./bin/Oracle/SqlTest.pl のように「./」で始まる必要があります。
User	文字列 - スカラー	内部データベースユーザー。詳細監視のためにデータベースに接続します。
Pword	文字列 - スカラー	内部データベースユーザー認証のための暗号化されたパスワード。 コマンドラインで入力する場合にのみパスワードを暗号化します。パスワードは、VCS Encrypt ユーティリティ (`/opt/VRTSvcs/bin/vcsencrypt`) を使って暗号化する必要があります。 **メモ:** この属性は PDB リソースには適用されません。
Table	文字列 - スカラー	`User/Pword` による更新用の表。
DBName	文字列 - スカラー	この属性は、データベースがポリシー管理の RAC データベースの場合にのみ設定します。この属性の値は一意のデータベース名に設定する必要があります。

オプション属性	データ形式と値のタイプ	定義
ManagedBy	文字列 - スカラー	この属性のデフォルト値は ADMIN です。ポリシー管理の RAC データベースでは、この属性は POLICY に設定する必要があります。
PDBName	文字列 - スカラー	PDB にこの属性を設定する必要があります。PDB データベース名に値を設定する必要があります。従来のデータベースおよび CDB データベースの場合、この属性を設定しないでください。
UseSystemD	ブール	systemD は、Linux オペレーティングシステム用のシステムおよびサービスマネージャです。systemD 機能をサポートする Linux の配布版全体でアプリケーションの管理を支援します。 online エントリポイントが、有効になった `UseSystemD` 属性で呼び出されると、Oracle リソースが system.slice 内のサービスとしてオンラインになります。`UseSystemD` 属性を有効にしない場合、通常の online エントリポイントが user.slice 内のリソースを開始します。 注意事項: システムの再ブート時に、アプリケーションリソースが user.slice 下で開始された場合、プロセスが適切に停止しない可能性があります。これにより、データベースが破損する場合があります。この動作は、Oracle アプリケーションのバージョンごとに異なります。このため、システム上で Oracle の UseSystemD 属性を有効にします。 アプリケーションプロセスは、`vcs-<resource_name>.service` (/etc/systemd/system 内) として表示できます。 `UseSystemD` 値を 0 または 1 に設定します。
SystemDAttrList	文字列 - スカラー	`UseSystemD` が有効になっていることを確認してから、キーと値のペアを `SystemDAttrList` 属性で設定します。詳しくは、Red Hat Web サイトを参照してアプリケーションに適用できる異なる制限を確認してください。 次に例を示します。 `# hares -modify oracle SystemDAttrList LimitLOCKS 44433` ここで、LimitLOCKS はキーで、44433 はキーの値です。

表 A-6は、Oracle エージェントの内部属性をリストします。この属性は内部使用のみを目的としています。 は、この属性の値を変更しないよう推奨します。

表 A-6　Oracle エージェントの内部属性

オプション属性	データ形式と値のタイプ	定義
AgentDirectory	静的 - 文字列	Oracle エージェントに関連付けられているバイナリやスクリプトなどのファイルの場所を指定します。 デフォルトは /opt/VRTSagents/ha/bin/Oracle です。

Netlsnr エージェント関数

リスナーとは、新規のクライアント接続要求を待機し、データベースに対するトラフィックを管理するサーバープロセスです。Netlsnr エージェントは、リスナーサービスをオンラインに切り替え、状態を監視し、オフラインに切り替えます。

表 A-7 に、Netlsnr エージェント関数を一覧で示します。

表 A-7　Netlsnr エージェント関数

エージェントの操作	説明
Online	次のコマンドを使ってリスナーのプロセスを開始します。 `lsnrctl start $LISTENER`
Offline	次のコマンドを使ってリスナーのプロセスを停止します。 `lsnrctl stop $LISTENER` パスワードを使ってリスナーが設定されている場合、エージェントは、そのパスワードを使ってそのリスナーを停止します。
Monitor	リスナープロセスの状態を確認します。 Netlsnr エージェントでは、基本と詳細の 2 段階の監視レベルを設定できます。 ■ 基本監視モードでは、エージェントは `tnslsnr` プロセスのプロセステーブルをスキャンして、リスナーのプロセスが実行中であることを確認します。 ■ 詳細監視モードでは、エージェントは `lsnrctl status $LISTENER` コマンドを使ってリスナープロセスの状態を確認します。(デフォルト)
Clean	`tnslsnr $LISTENER` のプロセステーブルをスキャンし、そのテーブルを破棄します。
Action	リソースで事前定義済みの処理を実行します。 p.456 の「Cluster Server agent for Oracle の Action エントリポイント」を参照してください。

Netlsnr エージェントのリソースタイプ定義

の Netlsnr エージェントは、Veritas InfoScale 製品 では Netlsnr リソースタイプとして表されます。

```
type Netlsnr (
 static str AgentDirectory = "/opt/VRTSagents/ha/bin/Netlsnr"
 static keylist SupportedActions = { VRTS_GetInstanceName,
        VRTS_GetRunningServices, "tnsadmin.vfd" }
 static str ArgList[] = { Owner, Home, TnsAdmin, Listener,
        EnvFile, MonScript, LsnrPwd, Encoding }
 static int IMF{} = { Mode=3, MonitorFreq=5, RegisterRetryLimit=3 }
 static str IMFRegList[] = { Home, Owner, Listener }
 str Owner
 str Home
 str TnsAdmin
 str Listener = "LISTENER"
 str EnvFile
 str MonScript = "./bin/Netlsnr/LsnrTest.pl"
 str LsnrPwd
 str Encoding
 static boolean IntentionalOffline = 0
)
```

Netlsnr エージェントの属性定義

Netlsnr エージェント属性の説明を確認します。エージェント属性は、必須、オプション、内部に分類されます。

表 A-8には、Netlsnr エージェントの必須属性がリストされています。必須属性には必ず値を割り当てる必要があります。

表 A-8 Netlsnr エージェントの必須属性

必須属性	データ形式と値のタイプ	定義
Owner	文字列 - スカラー	リスナープロセスを開始または停止する権限がある Oracle ユーザー。 エージェントでは、Oracle ユーザーとして LDAP ユーザーもサポートされます。
Home	文字列 - スカラー	Oracle バイナリファイルと設定ファイルへの $ORACLE_HOME パス。たとえば、パスとして /opt/ora_home を指定できます。 パスの最後にスラッシュ (/) は付けないでください。

表 A-9には、Netlsnr エージェントの必須属性がリストされています。オプションの属性は必要に応じて設定できます。

表 A-9 Netlsnr エージェントのオプションの属性

オプション属性	データ形式と値のタイプ	定義
TnsAdmin	文字列 - スカラー	Listener 設定ファイルが存在するディレクトリへの $TNS_ADMIN パス (listener.ora)。 デフォルトは /var/opt/oracle です。
Listener	文字列 - スカラー	Listener の名前。Netlsnr エージェントと Oracle データベースサーバーでは、Listener の名前の大文字小文字は区別されません。 デフォルトは LISTENER です。 **メモ:** グリッドインフラストラクチャが設定されている場合は、データベースリスナーにデフォルト名を使用しないようにしてください。そうしないと、グリッドインフラストラクチャがデフォルト名のリスナー (LISTENER) を起動し、さまざまな問題が発生する可能性があります。
LsnrPwd	文字列 - スカラー	リスナーの停止と監視に使用される Veritas InfoScale 製品 の暗号化されたパスワード。このパスワードは Listener 設定ファイルで設定されます。 コマンドラインで入力する場合にのみパスワードを暗号化します。パスワードは Veritas InfoScale 製品 Encrypt ユーティリティを使用して暗号化する必要があります。
EnvFile	文字列 - スカラー	エントリポイントスクリプトによって提供されるファイルの絶対パス名を指定します。このファイルには、Oracle リスナー環境のユーザーが設定する LD_LIBRARY_PATH などの環境変数が含まれます。 ファイル内容の構文は、Owner のログインシェルによって異なります。このファイルは Owner が読み取ることができる必要があります。ファイルにユーザー入力を求めるプロンプトが含まれていてはいけません。
MonScript	文字列 - スカラー	詳細な監視のために提供されているスクリプトへのパス名。デフォルトでは、リスナープロセスを監視する詳細監視は有効になっています。 **メモ:** 属性 MonScript の値が空の文字列に設定されている場合、エージェントは詳細監視を無効にします。 提供されている詳細監視スクリプトへのパス名は /opt/VRTSagents/ha/bin/Netlsnr/LsnrTest.pl です。 また、MonScript には /opt/VRTSagents/ha への相対パス名も指定できます。相対パス名は、パス ./bin/Netlsnr/LsnrTest.pl のように「./」で始まる必要があります。

表 A-10には、Netlsnr エージェントの内部属性がリストされています。この属性は内部使用のみを目的としています。は、この属性の値を変更しないよう推奨します。

表 A-10 Netlsnr エージェントの内部属性

オプション属性	データ形式と値のタイプ	定義
AgentDirectory	静的 - 文字列	Netlsnr エージェントに関連付けられているバイナリやスクリプトなどのファイルの場所を指定します。 デフォルトは /opt/VRTSagents/ha/bin/Netlsnr です。

B

クラスタ化された配備のサンプル設定ファイル

この付録では以下の項目について説明しています。

- サンプル設定ファイルについて
- Storage Foundation for High Availability のサンプル設定ファイル
- Storage Foundation for Clustered File System HA のサンプル設定ファイル
- Storage Foundation for Oracle RAC のサンプル設定ファイル

サンプル設定ファイルについて

このサンプル設定ファイルでは、いくつかの配備シナリオを説明します。

- Storage Foundation for High Availability
- Storage Foundation for Clustered File System HA
- Storage Foundation for Oracle RAC

環境を設定するためのガイドラインとしてサンプルファイルを使うことができます。

Oracle エージェントリソースが付属するサンプル設定ファイルは次のディレクトリにあります。

- SF HA の場合: /etc/VRTSagents/ha/conf/Oracle
- SFCFSHA の場合: /etc/VRTSagents/ha/conf/Oracle
- SF Oracle RAC の場合: /etc/VRTSvcs/conf/sample_rac

Storage Foundation for High Availability のサンプル設定ファイル

```
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster cluster1 (
        UserNames = { admin = anoGniNkoJooMwoInl }
        Administrators = { admin }
        )

system system1 (
        )

system system2 (
        )

group Oracle_Group (
        SystemList = { system1 = 0, system2 = 1 }
        AutoStartList = { system1 }
        )

        DiskGroup DG-orabindg (
                DiskGroup = qorasrc
                PanicSystemOnDGLoss = 1
                )

        DiskGroup DG-oradatadg (
                DiskGroup = dbdata_dg
                PanicSystemOnDGLoss = 1
                )

        IP Ora_IP (
                Device = NIC0
                Address = "10.182.187.108"
                NetMask = "255.255.248.0"
                )

        Mount ora_arch1_mnt (
```

```
        MountPoint = "/oraarch"
        BlockDevice = "/dev/vx/dsk/dbdata_dg/oraarchvol"
        FSType = vxfs
        FsckOpt = "-n"
        )

Mount ora_bin_mnt (
        MountPoint = "/opt/oracle/orahome"
        BlockDevice = "/dev/vx/dsk/qorasrc/orasrc10g"
        FSType = vxfs
        FsckOpt = "-n"
        )

Mount ora_data1_mnt (
        MountPoint = "/oradata"
        BlockDevice = "/dev/vx/dsk/dbdata_dg/oradatavol"
        FSType = vxfs
        FsckOpt = "-n"
        )

NIC Ora_NIC (
        Device = NIC0
        )

Oracle Ora_Oracle (
        Sid = ora10g
        Owner = oracle
        Home = "/opt/oracle/orahome"
        Pfile = "/opt/oracle/orahome/dbs/initora10g.ora"
        AgentDebug = 1
        )

Ora_IP requires Ora_NIC
Ora_Oracle requires ora_arch1_mnt
Ora_Oracle requires ora_bin_mnt
Ora_Oracle requires ora_data1_mnt
ora_arch1_mnt requires DG-oradatadg
ora_bin_mnt requires DG-orabindg
ora_data1_mnt requires DG-oradatadg


// resource dependency tree
//
```

```
//      group Oracle_Group
//      {
//      IP Ora_IP
//          {
//          NIC Ora_NIC
//          }
//      Oracle Ora_Oracle
//          {
//          Mount ora_arch1_mnt
//              {
//              DiskGroup DG-oradatadg
//              }
//          Mount ora_bin_mnt
//              {
//              DiskGroup DG-orabindg
//              }
//          Mount ora_data1_mnt
//              {
//              DiskGroup DG-oradatadg
//              }
//          }
//      }
```

Storage Foundation for Clustered File System HA のサンプル設定ファイル

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster cluster1 (
        UserNames = { admin = HopHojOlpKppNxpJom }
        Administrators = { admin }
        HacliUserLevel = COMMANDROOT
        )

system system1 (
        )
```

```
system system2 (
        )

group Oracle_Group (
        SystemList = { system1 = 0, system2 = 1 }
        AutoStartList = { system1 }

        )

        IP Ora_IP (
                Device = NIC0
                Address = "10.200.117.243"
                NetMask = "255.255.255.0"
                )

        NIC Ora_NIC (
                Device = NIC0
                )

        Oracle Ora_Oracle (
                Sid = ora11g
                Owner = oracle
                Home = "/opt/oracle/orahome"
                Pfile = "/opt/oracle/orahome/dbs/initora11g.ora"
                AgentDebug = 1
                )

        Ora_IP requires Ora_NIC
        Ora_Oracle requires Ora_IP


        // resource dependency tree
        //
        //      group Oracle_Group
        //      {
        //      Oracle Ora_Oracle
        //          {
        //          IP Ora_IP
        //              {
        //              NIC Ora_NIC
        //              }
        //          }
```

```
        //      }


group cvm (
        SystemList = { system1 = 0, system2 = 1 }
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { system1, system2 }
        )

        CFSMount oraarch_mnt (
                Critical = 0
                MountPoint = "/oraarch"
                BlockDevice = "/dev/vx/dsk/dbdata_dg/oraarchvol"
                )

        CFSMount orabin_mnt (
                Critical = 0
                MountPoint = "/ora11"
                BlockDevice = "/dev/vx/dsk/ora11bin/ora11vol"
                )

        CFSMount oradata2_mnt (
                Critical = 0
                MountPoint = "/oradata2"
                BlockDevice = "/dev/vx/dsk/oradatadg2/oradatavol2"
                )


        CFSMount dbdata_mnt (
                Critical = 0
                MountPoint = "/oradata"
                BlockDevice = "/dev/vx/dsk/dbdata_dg/oradatavol"
                )

        CFSMount oraredo_mnt (
                Critical = 0
                MountPoint = "/oraredo"
                BlockDevice = "/dev/vx/dsk/oradatadg2/oraredovol"
                )

        CFSfsckd vxfsckd (
                )
```

```
CVMCluster cvm_clus (
        CVMClustName = cluster1
        CVMNodeId = { system1 = 0, system2 = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )


CVMVolDg orabin_voldg (
        CVMDiskGroup = orabindg
        CVMVolume = { oracle_volume }
        CVMActivation = sw
        )

CVMVolDg oradata2_voldg (
        CVMDiskGroup = oradatadg2
        CVMVolume = { oradatavol2, oraredovol }
        CVMActivation = sw
        )

CVMVolDg dbdata_voldg (
        CVMDiskGroup = dbdata_dg
        CVMVolume = { oradatavol, oraarchvol }
        CVMActivation = sw
        )

CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

cvm_clus requires cvm_vxconfigd
oraarch_mnt requires dbdata_voldg
oraarch_mnt requires vxfsckd
orabin_mnt requires vxfsckd
orabin_voldg requires cvm_clus
oradata2_mnt requires oradata2_voldg
oradata2_mnt requires vxfsckd
oradata2_voldg requires cvm_clus
dbdata_mnt requires dbdata_voldg
dbdata_mnt requires vxfsckd
dbdata_voldg requires cvm_clus
```

```
oraredo_mnt requires oradata2_voldg
oraredo_mnt requires vxfsckd
vxfsckd requires cvm_clus


// resource dependency tree
//
//      group cvm
//      {
//      CFSMount oraarch_mnt
//          {
//          CVMVolDg dbdata_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CFSMount orabin_mnt
//          {
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CVMVolDg orabin_voldg
//          {
//          CVMCluster cvm_clus
//              {
//              CVMVxconfigd cvm_vxconfigd
//              }
//          }
```

```
//      CFSMount oradata2_mnt
//          {
//          CVMVolDg oradata2_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CVMVolDg orabin_voldg
//          {
//          CVMCluster cvm_clus
//              {
//              CVMVxconfigd cvm_vxconfigd
//              }
//          }
//      CFSMount oradata2_mnt
//          {
//          CVMVolDg oradata2_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CFSMount dbdata_mnt
//          {
//          CVMVolDg dbdata_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
```

```
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      CFSMount oraredo_mnt
//          {
//          CVMVolDg oradata2_voldg
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          CFSfsckd vxfsckd
//              {
//              CVMCluster cvm_clus
//                  {
//                  CVMVxconfigd cvm_vxconfigd
//                  }
//              }
//          }
//      }
```

Storage Foundation for Oracle RAC のサンプル設定ファイル

```
include "OracleASMTypes.cf"
include "types.cf"
include "CFSTypes.cf"
include "CRSResource.cf"
include "CVMTypes.cf"
include "MultiPrivNIC.cf"
include "OracleTypes.cf"
include "PrivNIC.cf"

cluster cluster1 (
        UserNames = { admin = hqrJqlQnrMrrPzrLqo,
                 "root@sxsvm02" = 0,
```

```
                 "root@sxsvm03" = 0 }
        Administrators = { admin, "root@sxsvm02",
                 "root@sxsvm03" }
        SecureClus = 1
        UseFence = SCSI3
        HacliUserLevel = COMMANDROOT
        )

system system1 (
        )

system system2 (
        )

group VxSS (
        SystemList = { system1 = 0, system2 = 1 }
        Parallel = 1
        AutoStartList = { system1, system2 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

        Phantom phantom_vxss (
                )

        ProcessOnOnly vxatd (
                IgnoreArgs = 1
                PathName = "/opt/VRTSat/bin/vxatd"
                )



        // resource dependency tree
        //
        //      group VxSS
        //      {
        //      Phantom phantom_vxss
        //      ProcessOnOnly vxatd
        //      }


group cvm (
        SystemList = { system1 = 0, system2 = 1 }
```

```
AutoFailOver = 0
Parallel = 1
AutoStartList = { system1, system2 }
)

Application cssd (
        Critical = 0
        StartProgram = "/opt/VRTSvcs/rac/bin/cssd-online"
        StopProgram = "/opt/VRTSvcs/rac/bin/cssd-offline"
        CleanProgram = "/opt/VRTSvcs/rac/bin/cssd-clean"
        MonitorProgram = "/opt/VRTSvcs/rac/bin/cssd-monitor"

        )

CFSMount ocrmnt (
        Critical = 0
        MountPoint = "/ocr_vote"
        BlockDevice = "/dev/vx/dsk/ocrdg/ocrvol"
        )

CFSfsckd vxfsckd (
        )

CVMCluster cvm_clus (
        CVMClustName = sfrac_rp2
        CVMNodeId = { system1 = 0, system2 = 1 }
        CVMTransport = gab
        CVMTimeout = 200
        )

CVMVolDg ocrvoldg (
        Critical = 0
        CVMDiskGroup = ocrdg
        CVMVolume = { ocrvol }
        CVMActivation = sw
        )

CVMVxconfigd cvm_vxconfigd (
        Critical = 0
        CVMVxconfigdArgs = { syslog }
        )

PrivNIC ora_priv (
```

```
        Critical = 0
        Device @system1 = { NIC1 = 0, NIC2 = 1 }
        Device @system2 = { NIC1 = 0, NIC2 = 1 }
        Address @system1 = "192.168.12.1"
        Address @system2 = "192.168.12.2"
        NetMask = "255.255.240.0"
        )

cssd requires ocrmnt
cssd requires ora_priv
cvm_clus requires cvm_vxconfigd
ocrmnt requires ocrvoldg
ocrmnt requires vxfsckd
ocrvoldg requires cvm_clus
vxfsckd requires cvm_clus


// resource dependency tree
//
//      group cvm
//      {
//      Application cssd
//          {
//          CFSMount ocrmnt
//              {
//              CVMVolDg ocrvoldg
//                  {
//                  CVMCluster cvm_clus
//                      {
//                      CVMVxconfigd cvm_vxconfigd
//                      }
//                  }
//              CFSfsckd vxfsckd
//                  {
//                  CVMCluster cvm_clus
//                      {
//                      CVMVxconfigd cvm_vxconfigd
//                      }
//                  }
//              }
//          PrivNIC ora_priv
//          }
//      }
```

```
group ora_db (
        SystemList = { system1 = 0, system2 = 1 }
        AutoStart = 0
        AutoFailOver = 0
        Parallel = 1
        AutoStartList = { system1, system2 }
        )

        CFSMount archive_mnt (
                Critical = 0
                MountPoint = "/oraarchive"
                BlockDevice = "/dev/vx/dsk/oradg/archivevol"
                )

        CFSMount ora_data_mnt (
                Critical = 0
                MountPoint = "/oradata"
                BlockDevice = "/dev/vx/dsk/oradg/oradatavol"
                )

        CVMVolDg ora_data_voldg (
                Critical = 0
                CVMDiskGroup = oradg
                CVMVolume = { oradatavol }
                CVMActivation = sw
                )

        Oracle ORACLE (
                Sid @system1 = orarp1
                Sid @system2 = orarp2
                Owner = oracle
                Home = "/oracle/app/orahome"
                Pfile @system1 = "/oradata/orarp/spfileorarp.ora"
                Pfile @system2 = "/oradata/orarp/spfileorarp.ora"
                StartUpOpt = SRVCTLSTART
                ShutDownOpt = SRVCTLSTOP
                )

        requires group cvm online local firm
        ORACLE requires archive_mnt
        ORACLE requires ora_data_mnt
```

```
archive_mnt requires ora_data_voldg
ora_data_mnt requires ora_data_voldg


// resource dependency tree
//
//      group ora_db
//      {
//      Oracle ORACLE
//          {
//          CFSMount archive_mnt
//              {
//              CVMVolDg ora_data_voldg
//              }
//          CFSMount ora_data_mnt
//              {
//              CVMVolDg ora_data_voldg
//              }
//          }
//      }
```

C

Database FlashSnap の状態情報

この付録では以下の項目について説明しています。

- Database FlashSnap の状態情報
- CLI からの Database FlashSnap スナップショットの状態情報

Database FlashSnap の状態情報

Veritas Database FlashSnap 機能により、様々なスナッププラン段階やスナップショット手順に関する次の情報が提供されます。

- スナップショットの状態情報
- スナップショットデータベースの状態情報

CLI からの Database FlashSnap スナップショットの状態情報

コマンドラインインターフェース(CLI)からスナップショット状態情報を表示するには、`-o list` オプションを指定した `dbed_vmchecksnap` コマンドを使い、指定したデータベースで利用可能なすべてのスナッププランを一覧表示します。スナップショット状態情報は、コマンド出力の `SNAP_STATUS` 列の下に表示されます。

メモ: スナップショット状態とスナップショットデータベース状態の情報はエラーメッセージにも表示されます。

CLI からのスナップショットの状態情報

表 C-1 に各スナップショット状態（SNAP_STATUS）値の詳しい情報を示します。

表 C-1　CLI からのスナップショットの状態情報

SNAP_STATUS	完了した操作	可能な操作
init_full	■ dbed_vmchecksnap -o validate（成功） ■ dbed_vmsnap -o resync（成功）	dbed_vmsnap -o snapshot
init_db	dbed_vmchecksnap -o validate -f *snapplan*（失敗）	ストレージ設定が正しく設定されていることを確認してください。
snapshot_start	dbed_vmsnap -o snapshot（失敗）	システム管理者に連絡してください。Veritas Volume Manager のコマンドを使ってスナップショットボリュームを再同期し、dbed_vmsnap-o snapshot -F で強制的にスナップショットを作成します。
snapshot_end	■ dbed_vmsnap -o snapshot（成功）	■ dbed_vmsnap -o resync ■ dbed_vmclonedb -o mount|mountdb|recoverdb
snapshot_vol_start snapshot_vol_end resync_dg_start resync_dg_end	dbed_vmsnap -o snapshot（失敗）	dbed_vmsnap -o snapshot を再実行します。
resync_vol_start resync_vol_end snapshot_dg_start snapshot_dg_end	dbed_vmsnap -o resync（失敗）	dbed_vmsnap -o resync を再実行します。

SNAP_STATUS	完了した操作	可能な操作
resync_start	dbed_vmsnap -o resync（失敗）	システム管理者に連絡してください。Veritas Volume Manager のコマンドを使ってスナップショットボリュームを再同期し、dbed_vmsnap-o snapshot -F で強制的にスナップショットを作成します。
mount start	dbed_vmclonedb -o mount（失敗）	dbed_vmclonedb -o umount
mount_end	dbed_vmclonedb -o mount （成功）	dbed_vmclonedb -o umount
restartdb_start	dbed_vmclonedb -o restartdb（失敗）	■ dbed_vmclonedb -o umount ■ スナップショットデータベースを手動で起動します。
restartdb_end	dbed_vmclonedb -o restartdb（成功）	dbed_vmclonedb -o umount
mountdb_start	dbed_vmclonedb -o mountdb（失敗）	スナップショットデータベースを手動でリカバリし、dbed_vmclonedb-o update_status を実行します。
mountdb_end	dbed_vmclonedb -o mountdb （成功）	■ dbed_vmclonedb -o update_status ■ dbed_vmclonedb -o umount
recoverdb_start	dbed_vmclonedb -o recoverdb （失敗）	■ スナップショットデータベースを手動でリカバリし、dbed_vmclonedb-o update_status を実行します。 ■ dbed_vmclonedb -o umount
recoverdb_end	dbed_vmclonedb -o recoverdb（成功）	dbed_vmclonedb -o umount

SNAP_STATUS	完了した操作	可能な操作
umount_start	dbed_vmclonedb -o umount(失敗)	ファイルシステムがビジー状態になっていないことを確認し、コマンドを再試行します。
umount_end	dbed_vmclonedb -o umount (成功)	■ dbed_vmclonedb -o mount ■ dbed_vmclonedb -o restartdb ■ dbed_vmsnap -o resync

CLI からのスナップショットデータベースの状態情報

コマンドラインからスナップショットデータベースの状態情報を表示するには、指定されたデータベースに使用可能なすべてのスナッププランを一覧表示する -o list オプションを指定して dbed_vmchecksnap コマンドを使います。スナップショットデータベースの状態情報は、列見出し DB_STATUS の下のコマンド出力に表示されます。

表 C-2 に各データベース状態(DB_STATUS)値の詳しい情報を示します。

表 C-2 CLI からのスナップショットデータベースの状態情報

DB_STATUS	完了した操作
init	■ dbed_vmchecksnap -o validate(成功) ■ dbed_vmsnap -o snapshot(成功)
database_recovered	dbed_vmclonedb -o recoverdb(成功)

D

サードパーティソフトウェアによるファイルのバックアップ

この付録では以下の項目について説明しています。

- サードパーティソフトウェアを使ったファイルのバックアップについて
- サードパーティソフトウェアによるファイルのバックアップ

サードパーティソフトウェアを使ったファイルのバックアップについて

SF Enterprise 製品 は、サードパーティソフトウェアを使用したファイルのバックアップをサポートしています。ただし、Quick I/O は SF Enterprise 製品 ではサポートされません。

サードパーティソフトウェアによるファイルのバックアップ

Veritas NetBackup 以外のサードパーティ製のバックアップソフトウェアを使用している場合、VxFS のエクステント属性をバックアップおよびリストアできることを確認してください。これは、リストアされた ODM（Oracle Disk Manager）のファイルが最適なパフォーマンスを得るために適切なエクステントのレイアウトに依存しているため、重要です。

Oracle RMAN を使った Oracle Disk Manager ファイルのバックアップとリストア

Oracle では、データベースのパフォーマンスを良好に保つために、連続したエクステントレイアウトを持つ Oracle Disk Manager ファイルを割り当てます。データベースファイ

ルをリストアするときには、それらのエクステントの属性を使ってファイルが割り当てられます。バックアップソフトウェアで従来の Oracle RMAN のバックアップ方式を使っている場合は、データファイルも正しいエクステントのレイアウトでリストアされます。

Veritas NetBackup 以外のバックアップソフトウェアで RMAN の「プロキシコピー」バックアップ方式を使っている場合は、エクステントの属性のバックアップが作成されない可能性があります。リストアされたデータファイルのエクステントのレイアウトが正しくなるように、odmmkfile コマンドを使って、失われたデータファイルを事前割り当てします。このコマンドは、ファイルをリストアする前にファイル用の連続した領域を事前割り当てします。

たとえば、100 M のサイズの Oracle データファイルを事前割り当てするには、Oracle データブロックサイズが 8K とすると、odmmkfile コマンドを使って次のように入力します。

```
# /opt/VRTS/bin/odmmkfile -h 8k -s 100m filename
```

odmmkfile コマンドについて詳しくは、odmmkfile(1) のマニュアルページを参照してください。