



# Oracleデータベースの移行

## Enterprise applications

NetApp  
February 10, 2026

# 目次

Oracleデータベースの移行 .....	1
概要 .....	1
スクリプト .....	1
移行計画 .....	1
計画に関する考慮事項 .....	2
データサイズ .....	2
データベース数 .....	2
リアーキテクチャノヨウケン .....	3
カットオーバー時間 .....	3
バックアウトパス .....	3
リハーサル .....	4
の手順 .....	4
概要 .....	4
データファイルの移行 .....	12
ログ配布 .....	12
ホストデータのコピー .....	48
ForeignLUNImport .....	85
サンプルスクリプト .....	106
データベースのシャットダウン .....	106
データベースの起動 .....	107
ファイルシステムを読み取り専用に変換 .....	108
ファイルシステムの交換 .....	109
データベース移行の自動化 .....	111
ファイルの場所を表示する .....	113
ASM移行のクリーンアップ .....	114
ASMからファイルシステム名への変換 .....	116
データベースでログを再生 .....	118
スタンバイデータベースでログを再生 .....	118

# Oracleデータベースの移行

## 概要

新しいストレージプラットフォームの機能を活用するには、必ず新しいストレージシステムにデータを配置する必要があるという、避けられない要件が1つあります。ONTAPを使用すると、ONTAPからONTAPへの移行とアップグレード、外部LUNのインポート、ホストオペレーティングシステムまたはOracleデータベースソフトウェアを直接使用する手順など、移行プロセスを簡単に実行できます。



以前に公開されていたテクニカルレポート\_TR-4534：『Migration of Oracle Databases to NetApp Storage Systems\_

新しいデータベースプロジェクトの場合、データベースとアプリケーション環境が適切に構築されているため、これは問題になりません。ただし、移行には、ビジネスの中断、移行の完了に必要な時間、必要なスキルセット、リスクの最小化という特別な課題が伴います。

## スクリプト

このドキュメントには、サンプルスクリプトが記載されています。これらのスクリプトは、ユーザによるミスの可能性を減らすために、移行のさまざまな側面を自動化するサンプル方法を提供します。スクリプトを使用すると、移行を担当するITスタッフの全体的な要求を軽減し、プロセス全体をスピードアップできます。これらのスクリプトはすべて、NetAppプロフェッショナルサービスとNetAppパートナーが実施した実際の移行プロジェクトを基に作成されています。このドキュメント全体で使用例が示されています。

## 移行計画

Oracleのデータ移行は、データベース、ホスト、ストレージアレイの3つのレベルのいずれかで実行できます。

違いは、解決策全体のどのコンポーネント（データベース、ホストオペレーティングシステム、ストレージシステム）がデータの移動を担当しているかです。

次の図は、移行レベルとデータフローの例を示しています。データベースレベルの移行では、元のストレージシステムからホストレイヤとデータベースレイヤを経由して新しい環境にデータが移動されます。ホストレベルの移行も同様ですが、データはアプリケーションレイヤを通過せず、代わりにホストプロセスを使用して新しい場所に書き込まれます。最後に、ストレージレベルの移行では、NetApp FASシステムなどのアレイがデータ移動を行います。

[エラー：グラフィックイメージがありません]

データベースレベルの移行とは、通常、スタンバイデータベースを介してOracleのログ配布を使用し、Oracleレイヤでの移行を完了することを指します。ホストレベルの移行は、ホストオペレーティングシステム構成の標準機能を使用して実行されます。この構成には、cp、tar、およびOracle Recovery Manager (RMAN) などのコマンドを使用するか、論理ボリュームマネージャ (LVM) を使用してファイルシステムの基盤となるバイトを再配置するファイルコピー処理が含まれます。Oracle Automatic Storage Management (ASM) は、データベースアプリケーションのレベル以下で実行されるため、ホストレベルの機能に分類されます。ASMは、ホスト上の通常の論理ボリュームマネージャの代わりに使用されます。最後に、データを

ストレージアレイレベル（オペレーティングシステムのレベル以下）で移行できます。

## 計画に関する考慮事項

移行に最適な方法は、移行する環境の規模、ダウンタイムの回避の必要性、移行の実行に必要な全体的な作業など、さまざまな要因の組み合わせによって異なります。大規模なデータベースの移行には明らかに多くの時間と労力が必要ですが、そのような移行の複雑さは最小限です。小規模なデータベースは迅速に移行できますが、数千ものデータベースを移行する必要がある場合は、規模の大きさによって複雑な作業が発生する可能性があります。最後に、データベースの規模が大きいほど、ビジネスクリティカルである可能性が高くなります。そのため、バックアウトパスを維持しながらダウンタイムを最小限に抑える必要があります。

ここでは、移行戦略を計画する際の考慮事項について説明します。

## データサイズ

移動するデータベースのサイズは移行計画に明らかに影響しますが、サイズがカットオーバー時間に必ずしも影響するとは限りません。大量のデータを移行する必要がある場合、主に帯域幅を考慮する必要があります。コピー処理は通常、効率的なシーケンシャルI/Oを使用して実行されます。控えめな見積もりでは、コピー処理に使用できるネットワーク帯域幅の使用率が50%であると想定しています。たとえば、8GBのFCポートでは理論上約800MBpsの転送が可能です。使用率が50%であれば、約400Mbpsの速度でデータベースをコピーできます。そのため、この速度では約7時間で10TBのデータベースをコピーできます。

長距離の移行では、通常、ログ配布プロセスなど、より創造的なアプローチが必要になります。詳細については、を参照してください。 ["データファイルのオンライン移動"](#)。長距離IPネットワークでは、LANやSANの速度に近い場所で帯域幅が使用されることはほとんどありません。あるケースでは、アーカイブログの生成頻度が非常に高い220TBデータベースの長距離移行をNetAppが支援しました。データ転送のために選択されたアプローチは、可能な限り最大の帯域幅を提供するため、テープの毎日の出荷でした。

## データベース数

多くの場合、大量のデータを移動する際の問題はデータサイズではなく、データベースをサポートする構成の複雑さです。50TBのデータベースを移行する必要があるというだけでは、十分な情報ではありません。1つの50TBのミッションクリティカルなデータベース、4,000個のレガシーデータベースの集まり、または本番環境と非本番環境の混在環境などが考えられます。場合によっては、ほとんどのデータがソースデータベースのクローンで構成されています。これらのクローンは簡単に再作成できるため、マイグレートする必要はありません。特に、新しいアーキテクチャでNetApp FlexCloneボリュームを利用するように設計されている場合は、クローンを簡単に再作成できるためです。

移行を計画するには、対象となるデータベースの数と、データベースの優先順位を決定する方法を理解しておく必要があります。データベースの数が増えるにつれて、優先される移行オプションはスタック内で徐々に低くなる傾向があります。たとえば、RMANを使用して単一のデータベースのコピーを簡単に実行でき、短時間の停止が発生する可能性があります。これはホストレベルのレプリケーションです。

データベースが50個ある場合は、RMANコピーを受信する新しいファイルシステム構造を設定してデータを所定の場所に移動することを避けた方が簡単な場合があります。このプロセスは、ホストベースのLVM移行を利用して、古いLUNから新しいLUNにデータを再配置することで実行できます。これにより、データベース管理者（DBA）チームからOSチームに責任が移され、データベースに関して透過的にデータが移行されます。ファイルシステム構成は変更されません。

最後に、200台のサーバにまたがる500個のデータベースを移行する必要がある場合は、ONTAP Foreign LUN Import (FLI) 機能などのストレージベースのオプションを使用して、LUNの直接移行を実行できます。

## リアーキテクチャノユウケン

通常、データベースファイルのレイアウトを変更して新しいストレージレイの機能を利用する必要がありますが、必ずしもそうであるとは限りません。たとえば、EFシリーズオールフラッシュレイの機能は、主にSANのパフォーマンスと信頼性を重視しています。ほとんどの場合、データレイアウトに関する特別な考慮事項なしに、データベースをEFシリーズレイに移行できます。要件は、高いIOPS、低レイテンシ、堅牢な信頼性だけです。RAID構成やDynamic Disk Poolsなどの要素に関連するベストプラクティスはありますが、EFシリーズのプロジェクトでこれらの機能を活用するためにストレージアーキテクチャ全体を大幅に変更しなければならないことはほとんどありません。

これとは対照的に、ONTAPへの移行では、最終的な構成で最大限の価値を実現するために、データベースレイアウトをより慎重に検討する必要があります。ONTAP自体は、特定のアーキテクチャの作業がなくても、データベース環境に多くの機能を提供します。最も重要なのは、現在のハードウェアが寿命に達したときに、システムを停止せずに新しいハードウェアに移行できることです。一般的には、ONTAPへの移行は最後に実行する必要があります。後続のハードウェアはインプレースでアップグレードされ、データは無停止で新しいメディアに移行されます。

いくつかの計画では、さらに多くの利点が利用可能です。スナップショットの使用には、最も重要な考慮事項があります。Snapshotは、バックアップ、リストア、クローニング処理をほぼ瞬時に実行するための基盤です。スナップショットの機能の一例として、最大の用途は、6台のコントローラ上の約250個のLUNで996TBの単一データベースを実行している場合です。このデータベースのバックアップは2分で完了し、リストアは2分で完了し、クローニングは15分で完了します。その他のメリットとしては、ワークロードの変化に応じてクラスタ内でデータを移動できる機能や、サービス品質（QoS）制御を適用して、マルチデータベース環境で安定した優れたパフォーマンスを提供できる機能などがあります。

QoS制御、データ再配置、Snapshot、クローニングなどのテクノロジーは、ほぼすべての構成で機能します。しかし、利益を最大化するためには、一般的にいくつかの考えが必要です。場合によっては、新しいストレージレイへの投資を最大限に活用するために、データベースストレージのレイアウトの設計変更が必要になることがあります。ホストベースまたはストレージベースの移行では元のデータレイアウトがレプリケートされるため、このような設計の変更は移行戦略に影響する可能性があります。移行を完了してONTAP向けに最適化されたデータレイアウトを提供するには、追加の手順が必要になる場合があります。に示す手順 ["Oracle移行手順の概要"](#) また、データベースを移行するだけでなく、最小限の労力で最適な最終レイアウトに移行する方法についても説明します。

## カットオーバー時間

カットオーバー中のサービス停止の許容最大値を決定する必要があります。移行プロセス全体がシステム停止を引き起こすと想定するのはよくある間違いです。サービスの中断が発生する前に多くのタスクを完了できます。また、多くのオプションを使用すると、システム停止やシステム停止を伴わずに移行を完了できます。カットオーバーの時間は手順によって異なるため、システム停止が避けられない場合でも、許容されるサービス停止の最大値を定義する必要があります手順。

たとえば、10TBのデータベースのコピーには、通常、約7時間かかります。ビジネスニーズが7時間の停止を許容している場合、ファイルのコピーは簡単で安全な移行オプションです。5時間に対応できない場合は、シンプルなログ配布プロセス（["Oracleのログ配布"](#)）最小限の労力でセットアップでき、カットオーバー時間を約15分に短縮できます。この間、データベース管理者はプロセスを完了できます。許容できない時間が15分であった場合は、スクリプトを使用して最終カットオーバープロセスを自動化し、カットオーバー時間をわずか数分に短縮できます。移行はいつでも高速化できますが、そのためには時間と労力がかかります。カットオーバーの所要時間は、ビジネス部門が許容できる範囲で決定する必要があります。

## バックアウトパス

完全にリスクのない移行はありません。テクノロジーが完全に動作していても、ユーザエラーの可能性は常にあ

ります。選択した移行パスに関連するリスクと、失敗した移行の結果を考慮する必要があります。たとえば、Oracle ASMの透過的オンラインストレージ移行機能は、Oracle ASMの主要機能の1つであり、この方法は、最も信頼性の高い方法の1つです。ただし、この方法ではデータが不可逆的にコピーされています。万一ASMで問題が発生した場合、簡単にバックアウトできるパスはありません。唯一の選択肢は、元の環境をリストアするか、ASMを使用して移行を元のLUNに戻すことです。このリスクは、元のストレージ・システムでスナップショット・タイプのバックアップを実行できる場合には、最小限に抑えることができますが、排除することはできません。

## リハーサル

一部の移行手順は、実行前に完全に検証する必要があります。移行とカットオーバープロセスのリハーサルは、ミッションクリティカルなデータベースへの一般的な要求であり、移行を成功させ、ダウンタイムを最小限に抑える必要があります。また、ユーザ受け入れテストは移行後の作業に含まれることが多く、システム全体を本番環境に戻すには、これらのテストが完了する必要があります。

リハーサルが必要な場合は、いくつかのONTAP機能を使用すると、プロセスがはるかに簡単になります。特に、スナップショットを使用すると、テスト環境をリセットして、データベース環境のスペース効率に優れた複数のコピーをすばやく作成できます。

## の手順

### 概要

Oracleデータベースへの移行には、さまざまな手順を使用できます。最適なソリューションは、ビジネスニーズに応じて異なります。

多くの場合、システム管理者とDBAには、物理ボリュームのデータの再配置、ミラーリングとミラーリングの解除、Oracle RMANを使用したデータのコピーなど、独自の方法があります。

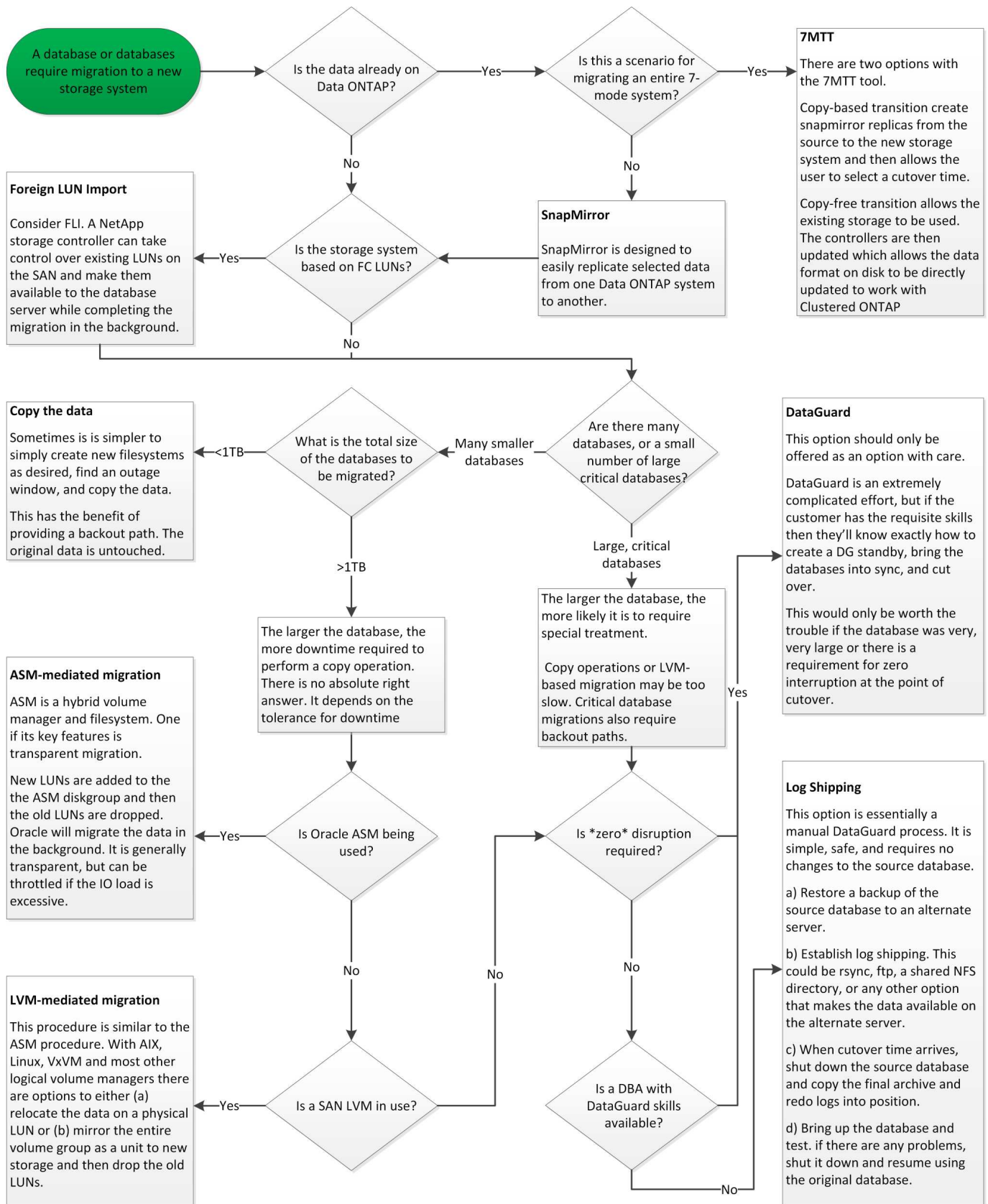
これらの手順は、主に、利用可能なオプションの一部に慣れていないITスタッフ向けのガイダンスとして提供されています。さらに、各移行アプローチのタスク、時間要件、スキルセットの要求についても説明します。これにより、NetAppやパートナーのプロフェッショナルサービスやIT管理者などの他の関係者が、各手順の要件をより十分に理解できるようになります。

移行戦略を作成するための単一のベストプラクティスはありません。計画を作成するには、まず可用性オプションを理解してから、ビジネスのニーズに最適な方法を選択する必要があります。次の図は、基本的な考慮事項と一般的な結論を示していますが、すべての状況に当てはまるわけではありません。

たとえば、1つのステップで合計データベースサイズの問題が生成されます。次の手順は、データベースが1TBを超えているかどうかによって異なります。推奨される手順は、一般的なお客様の慣行に基づいた推奨事項です。ほとんどのお客様は、DataGuardを使用して小規模なデータベースをコピーすることはありませんが、場合によってはコピーすることもあります。ほとんどのお客様は、時間がかかるため50TBのデータベースをコピーしようとはしませんが、一部のお客様では、このような処理を実行できるだけの十分なメンテナンス時間がある場合があります。

次のフローチャートは、移行パスが最適な考慮事項のタイプを示しています。画像を右クリックして新しいタブで開くと、読みやすさが向上します。





です。

## データファイルのオンライン移動

Oracle 12cR1以降では、データベースをオンラインにしたままデータファイルを移動できます。さらに、異

なる種類のファイルシステム間で動作します。たとえば、データファイルをxfsファイルシステムからASMに再配置できます。個々のデータファイルの移動処理が必要になるため、この方法は一般に大規模な環境では使用されませんが、データファイルの数が少ない小規模なデータベースの場合は検討する価値があります。

また、データファイルを移動するだけで、既存のデータベースの一部を移行することもできます。たとえば、アクティブでないデータファイルをコスト効率に優れたストレージ（アイドルブロックをオブジェクトストアに格納できるFabricPoolなど）に再配置できます。

## データベースレベルの移行

データベースレベルでの移行とは、データベースがデータを再配置できることを意味します。具体的には、ログ配布を意味します。RMANやASMなどのテクノロジーはOracle製品ですが、移行の目的では、ファイルのコピーやボリュームの管理を行うホストレベルで動作します。

### ログ配布

データベースレベルの移行の基盤となるのがOracleアーカイブログです。このログには、データベースに対する変更のログが記録されます。ほとんどの場合、アーカイブログはバックアップおよびリカバリ戦略の一部です。リカバリプロセスでは、まずデータベースをリストアし、次に1つ以上のアーカイブログを再生して、データベースを目的の状態に戻します。これと同じ基本テクノロジーを使用して、運用の中断をほとんどまたはまったく伴わずに移行を実行できます。さらに重要なのは、このテクノロジーにより、元のデータベースに手を加えずに移行できるため、バックアウトパスが維持されます。

移行プロセスは、まずセカンダリサーバへのデータベースバックアップのリストアから始まります。これにはさまざまな方法がありますが、ほとんどのお客様は通常のバックアップアプリケーションを使用してデータファイルをリストアしています。データファイルがリストアされたら、ユーザはログの配布方法を設定します。その目的は、プライマリデータベースで生成されたアーカイブログの一定のフィードを作成し、リストアしたデータベースでそれらのログを再生して、両方を同じ状態に保つことです。カットオーバー時間に達すると、ソースデータベースが完全にシャットダウンされ、最終的なアーカイブログと場合によってはREDOログがコピーされて再生されます。コミットされた最終トランザクションの一部がREDOログに含まれている可能性があるため、REDOログも考慮することが重要です。

これらのログが転送されて再生されると、両方のデータベースの整合性が維持されます。この時点で、ほとんどのお客様はいくつかの基本的なテストを実施します。移行プロセス中にエラーが発生した場合は、ログ再生でエラーが報告されて失敗します。構成が最適であることを確認するために、既知のクエリまたはアプリケーションベースのアクティビティに基づいていくつかのクイックテストを実行することをお勧めします。また、移行したデータベースに元のデータベースが存在するかどうかを確認する前に、最後のテストテーブルを1つ作成してから元のデータベースをシャットダウンすることも一般的です。この手順では、最終的なログ同期中にエラーが発生していないことを確認します。

シンプルなログ配布移行は、元のデータベースに対してアウトオブバンドで構成できるため、ミッションクリティカルなデータベースに特に役立ちます。ソースデータベースの構成の変更は必要ありません。移行環境のリストアと初期設定は、本番環境の運用には影響しません。ログ配布を構成すると、一部のI/O要求が本番サーバに送信されます。ただし、ログ配布ではアーカイブログの単純なシーケンシャル読み取りが行われるため、本番データベースのパフォーマンスへの影響はほとんどありません。

ログ配布は、長距離の変更率の高い移行プロジェクトに特に有用であることが証明されています。たとえば、1つの220TBデータベースを約500マイル離れた場所に移行しました。変更率は非常に高く、セキュリティ上の制約があるため、ネットワーク接続を使用できませんでした。ログ配布はテープと宅配便を使用して実施しました。ソース・データベースのコピーは最初に以下の手順を使用してリストアされました。ログは、カットオーバーの最終セットが配信され、ログがレプリカデータベースに適用されるまで、宅配便によって毎週出荷されました。



場合によっては、完全なDataGuard環境が保証されることもあります。ログ配布またはスタンバイデータベースの構成をDataGuardと呼ぶのは正しくありません。Oracle DataGuardは、データベースレプリケーションを管理するための包括的なフレームワークですが、レプリケーションテクノロジーではありません。移行作業における完全なDataGuard環境の主なメリットは、データベース間で透過的にスイッチオーバーできることです。また、新しい環境とのパフォーマンスやネットワーク接続問題などの問題が検出された場合に、元のデータベースに透過的にスイッチオーバーすることもできます。DataGuard環境を完全に構成するには、データベースレイヤだけでなくアプリケーションも構成して、アプリケーションがプライマリデータベースの場所の変更を検出できるようにする必要があります。一般的に、DataGuardを使用して移行を完了する必要はありませんが、DataGuardに関する豊富な専門知識を社内に持ち、移行作業にすでにDataGuardを利用しているお客様もいます。

### 再アーキテクチャ

前述したように、ストレージレイの高度な機能を活用するには、データベースレイアウトの変更が必要になる場合があります。さらに、ASMからNFSファイルシステムへの移行などのストレージプロトコルの変更によって、ファイルシステムのレイアウトが変更される必要があります。

DataGuardなどのログ配布方法の主な利点の1つは、レプリケーション先がソースと一致している必要がないことです。ログ配布アプローチを使用してASMから通常のファイルシステムに（またはその逆に）移行する場合、問題はありません。データファイルの正確なレイアウトを宛先で変更して、Pluggable Database (PDB) テクノロジーの使用を最適化したり、特定のファイルに対してQoS制御を選択的に設定したりできます。つまり、ログ配布に基づく移行プロセスを使用すると、データベースストレージレイアウトを簡単かつ安全に最適化できます。

### サーバーリソース

データベースレベルの移行の制限事項の1つに、2台目のサーバの必要性があります。この2台目のサーバは、次の2つの方法で使用できます。

1. 2番目のサーバは、データベースの永続的な新しいホームとして使用できます。
2. 2番目のサーバを一時的なステージングサーバとして使用できます。新しいストレージレイへのデータ移行が完了してテストされると、LUNまたはNFSファイルシステムがステージングサーバから切断され、元のサーバに再接続されます。

最初のオプションは最も簡単ですが、非常に強力なサーバを必要とする非常に大規模な環境では使用できない可能性があります。2番目のオプションでは、ファイルシステムを元の場所に再配置するための追加作業が必要です。ファイルシステムをステージングサーバからアンマウントして元のサーバに再マウントできるため、NFSをストレージプロトコルとして使用する単純な操作にすることができます。

ブロックベースのファイルシステムでは、FCゾーニングまたはiSCSIイニシエータを更新するために追加の作業が必要です。ほとんどの論理ボリュームマネージャ（ASMを含む）では、元のサーバでLUNが使用可能になると、LUNが自動的に検出されてオンラインになります。ただし、ファイルシステムやLVMの実装によっては、データのエクスポートとインポートにより多くの作業が必要になる場合があります。正確な手順は異なる場合がありますが、通常は、移行を完了し、元のサーバにデータをリホームするためのシンプルで反復可能な手順を確立するのは簡単です。

単一のサーバ環境内でログ配布を設定してデータベースをレプリケートすることは可能ですが、ログを再生するには、新しいインスタンスに別のプロセスSIDを設定する必要があります。異なるSIDを持つ別のプロセスIDセットの下でデータベースを一時的に起動し、後で変更することができます。ただし、管理作業が複雑になり、データベース環境がユーザミスのリスクにさらされる可能性があります。

## ホストレベルの移行

ホストレベルでデータを移行するとは、ホストオペレーティングシステムと関連するユーティリティを使用して移行を完了することを意味します。このプロセスには、Oracle RMANやOracle ASMなど、データをコピーするすべてのユーティリティが含まれます。

### データコピー

単純なコピー操作の値を過小評価してはなりません。最新のネットワークインフラでは、1秒あたりのギガバイト数でデータを移動できます。ファイルのコピー処理は、効率的なシーケンシャル読み取り/書き込みI/Oに基づいています。ログ配布と比較すると、ホストのコピー処理ではこれ以上のシステム停止は避けられませんが、移行は単なるデータ移動ではありません。通常は、ネットワークへの変更、データベースの再起動時間、移行後のテストが含まれます。

データのコピーに実際に必要な時間はそれほど長くはありません。さらに、コピー処理では、元のデータが変更されないため、保証されたバックアウトパスが維持されます。移行プロセス中に問題が発生した場合は、元のデータを持つ元のファイルシステムを再アクティブ化できます。

### プラットフォームの変更

再プラットフォーム化とは、CPUタイプの変更を指します。従来のSolaris、AIX、またはHP-UXプラットフォームからx86 Linuxにデータベースを移行する場合、CPUアーキテクチャの変更により、データを再フォーマットする必要があります。SPARC、IA64、POWER CPUはビッグエンディアンプロセッサとして知られ、x86とx86\_64アーキテクチャはリトルエンディアンとして知られている。その結果、Oracleデータファイル内の一部のデータは、使用中のプロセッサによって順序が異なります。

従来、お客様はDataPumpを使用してプラットフォーム間でデータをレプリケートしてきました。データダンプは、ターゲットデータベースでより迅速にインポートできる特別なタイプの論理データエクスポートを作成するユーティリティです。データの論理コピーが作成されるため、DataPumpはプロセッサエンディアンの依存関係を残します。一部のお客様はデータダンプを再プラットフォーム化に使用していますが、Oracle 11gではより高速なオプションが利用できるようになりました。クロスプラットフォームで移動可能な表領域です。このアドバンスにより、テーブルスペースを別のエンディアン形式に変換できます。これは、DataPumpエクスポートよりも優れたパフォーマンスを提供する物理的な変換です。DataPumpエクスポートでは、物理バイトを論理データに変換してから、物理バイトに戻す必要があります。

DataPumpと移動可能な表領域の詳細については、NetAppのドキュメントでは説明していませんが、NetAppでは、新しいCPUアーキテクチャを使用して新しいストレージアレイログに移行する際にお客様をサポートしてきた経験に基づいて、次のような推奨事項がいくつかあります。

- DataPumpを使用している場合は、移行の完了に必要な時間をテスト環境で測定する必要があります。お客様は、移行の完了に必要な時間に驚かれることがあります。このような予期しないダウンタイムが発生すると、原因の停止が発生
- 多くのお客様は、クロスプラットフォームの移動可能な表領域はデータ変換を必要としないと誤って考えています。異なるエンディアンを持つCPUが使用されている場合、RMAN convert データファイルに対しては、事前に操作を実行しておく必要があります。これは瞬間的な操作ではありません。場合によっては、異なるデータファイルで複数のスレッドを動作させることで変換処理を高速化することができますが、変換処理を回避することはできません。

### 論理ボリュームマネージャによる移行

LVMは、1つ以上のLUNのグループを作成し、それらをエクステントと呼ばれる小さな単位に分割することで機能します。次に、エクステントのプールをソースとして使用して、基本的に仮想化された論理ボリュームを作成します。この仮想化レイヤーは、さまざまな方法で価値を提供します。

- 論理ボリュームは、複数のLUNから取得されたエクステントを使用できます。論理ボリューム上に作成されたファイルシステムは、すべてのLUNのパフォーマンス機能をフルに使用できます。また、ボリュームグループ内のすべてのLUNの均等なロードが促進され、より予測可能なパフォーマンスが提供されます。
- 論理ボリュームのサイズは、エクステントを追加したり、場合によっては削除したりすることで変更できます。論理ボリューム上のファイルシステムのサイズ変更は、通常無停止で実行されます。
- 基盤となるエクステントを移動することで、論理ボリュームを無停止で移行できます。

LVMを使用した移行は、エクステントの移動またはエクステントのミラーリング/ミラーリングという2つの方法のいずれかで機能します。LVMの移行では、効率的な大容量ブロックのシーケンシャルI/Oが使用され、パフォーマンスに関する懸念が生じることはほとんどありません。これが問題になった場合は、通常、I/O速度を調整するオプションがあります。これにより、移行の完了に必要な時間が長くなりますが、ホストとストレージシステムのI/O負荷が軽減されます。

## ミラーおよびデミラー

AIX LVMなどの一部のボリュームマネージャでは、各エクステントのコピー数を指定したり、各コピーをホストするデバイスを制御したりできます。移行では、既存の論理ボリュームを取得し、基盤となるエクステントを新しいボリュームにミラーリングし、コピーの同期を待ってから、古いコピーをドロップします。バックアップが必要な場合は、ミラーコピーが破棄される前に元のデータのSnapshotを作成できます。または、サーバを短時間シャットダウンして元のLUNをマスクしてから、格納されているミラーコピーを強制的に削除することもできます。これにより、リカバリ可能なデータのコピーが元の場所に保持されます。

## エクステントの移行

ほとんどすべてのボリューム・マネージャではエクステントの移行が可能であり、複数のオプションが存在する場合もあります。たとえば、一部のボリュームマネージャでは、管理者が特定の論理ボリュームの個々のエクステントを古いストレージから新しいストレージに再配置できます。Linux LVM2などのボリュームマネージャは、`pvmove` コマンド。指定したLUNデバイス上のすべてのエクステントを新しいLUNに再配置します。古いLUNは退避後に削除できます。



運用の主なリスクは、古い未使用のLUNを構成から削除することです。FCゾーニングを変更したり、古いLUNデバイスを削除したりする場合は、十分に注意する必要があります。

## Oracle自動ストレージ管理

Oracle ASMは、論理ボリュームマネージャとファイルシステムを組み合わせたものです。大まかに言えば、Oracle ASMはLUNの集まりを受け取り、それらを小さな割り当て単位に分割して、ASMディスクグループと呼ばれる単一のボリュームとして提供します。ASMには、冗長性レベルを設定してディスクグループをミラーリングする機能もあります。ボリュームは、ミラーリングされていない（外部冗長性）、ミラーリングされている（通常の冗長性）、または3方向ミラーリングされている（高冗長性）ことができます。冗長性レベルの設定は作成後に変更できないため、慎重に行う必要があります。

ASMは、ファイルシステム機能も提供します。ファイルシステムはホストから直接認識されませんが、OracleデータベースではASMディスクグループ上のファイルやディレクトリを作成、移動、削除できます。また、`asmcmd`ユーティリティを使用して構造体をナビゲートすることもできます。

他のLVM実装と同様に、Oracle ASMは、使用可能なすべてのLUNにわたって各ファイルのI/Oをストライピングおよびロードバランシングすることで、I/Oパフォーマンスを最適化します。次に、基盤となるエクステントを再配置して、ASMディスクグループのサイズ変更と移行の両方を可能にします。Oracle ASMは、リバランシング処理を通じてプロセスを自動化します。新しいLUNがASMディスクグループに追加され、古いLUNが削除されると、エクステントの再配置と、退避したLUNがディスクグループから削除されます。このプロセスは、最も実証された移行方法の1つであり、透過的な移行を提供するASMの信頼性は、ASMの最も重要な機

能である可能性があります。



Oracle ASMのミラーリングレベルは固定されているため、mirrorおよびdemirror方式の移行では使用できません。

## ストレージレベルの移行

ストレージレベルの移行とは、アプリケーションレベルとオペレーティングシステムレベルの両方を下回るレベルで移行を実行することを意味します。以前は、これはネットワークレベルでLUNをコピーする専用のデバイスを使用することを意味していましたが、現在ではこれらの機能はONTAPに標準で搭載されています。

### SnapMirror

NetAppシステム間でのデータベースの移行は、ほとんどの場合、NetApp SnapMirrorデータレプリケーションソフトウェアを使用して実行されます。このプロセスでは、移動するボリュームのミラー関係を設定して同期を許可し、カットオーバー時間を待機します。到着すると、ソースデータベースがシャットダウンされ、最後のミラー更新が1回実行され、ミラーが解除されます。レプリカボリュームは、格納されているNFSファイルシステムディレクトリをマウントするか、格納されているLUNを検出してデータベースを開始することで、使用できる状態になります。

単一のONTAPクラスタ内でのボリュームの再配置は、移動とはみなされず、日常的な作業です。volume move 操作。SnapMirrorは、クラスタ内でデータレプリケーションエンジンとして使用されます。このプロセスは完全に自動化されています。LUNマッピングやNFSエクスポート権限など、ボリュームの属性がボリューム自体と一緒に移動された場合に実行する追加の移行手順はありません。再配置では、ホストの処理が中断されません。場合によっては、再配置されたデータに可能な限り効率的にアクセスできるようにネットワークアクセスを更新する必要がありますが、これらのタスクも無停止で実行できます。

### Foreign LUN Import (FLI)

FLIは、8.3以降を実行するData ONTAPシステムで既存のLUNを別のストレージアレイから移行できる機能です。手順はシンプルです。ONTAPシステムは、他のSANホストと同様に既存のストレージアレイにゾーニングされます。次に、Data ONTAPが必要な従来型LUNを制御し、基盤となるデータを移行します。また、インポートプロセスでは、データの移動時に新しいボリュームの効率化設定が使用されます。つまり、移動プロセス中にデータをインラインで圧縮したり重複排除したりできます。

Data ONTAP 8.3で初めて実装されたFLIでは、オフライン移行のみが可能でした。これは非常に高速な転送でしたが、移行が完了するまでLUNデータを使用できないことを意味していました。オンライン移行はData ONTAP 8.3.1で導入されました。このような移行では、転送プロセス中にONTAPがLUNデータを提供できるようになるため、システム停止を最小限に抑えることができます。ONTAP経由でLUNを使用するようにホストをゾーニングしている間、システムが短時間停止します。ただし、これらの変更が行われるとすぐに、データに再びアクセスでき、移行プロセス中も引き続きアクセスできます。

コピー処理が完了するまで読み取りI/OはONTAP経由でプロキシされ、書き込みI/Oは外部LUNとONTAP LUNの両方に同期的に書き込まれます。管理者が完全なカットオーバーを実行して外部LUNを解放し、書き込みをレプリケートしなくなるまで、2つのLUNコピーはこの方法で同期されます。

FLIはFCと連携するように設計されていますが、iSCSIに変更する必要がある場合は、移行の完了後に、移行したLUNをiSCSI LUNとして簡単に再マッピングできます。

FLIの機能の1つに、アライメントの自動検出と調整があります。アライメントという用語は、LUNデバイス上のパーティションを指します。パフォーマンスを最適化するには、I/Oが4Kブロックにアライメントされている必要があります。パーティションを4Kの倍数ではないオフセットに配置すると、パフォーマンスが低下します。

アライメントには、パーティションオフセット（ファイルシステムのブロックサイズ）を調整して修正できないもう1つの側面があります。たとえば、ZFSファイルシステムのデフォルトの内部ブロックサイズは512バイトです。AIXを使用しているお客様の中には、ブロックサイズが512バイトまたは1バイトのJFS2ファイルシステムを作成するケースもあります。ファイルシステムは4Kの境界にアライメントされていても、そのファイルシステム内に作成されたファイルはアライメントされず、パフォーマンスが低下します。

このような状況ではFLIを使用しないでください。移行後はデータにアクセスできますが、その結果、ファイルシステムのパフォーマンスが大幅に制限されます。一般的な原則として、ONTAPでランダムオーバーライトワークロードをサポートするファイルシステムでは、4Kブロックサイズを使用する必要があります。これは主に、データベースデータファイルやVDI環境などのワークロードに該当します。ブロックサイズは、関連するホストオペレーティングシステムコマンドを使用して特定できます。

たとえば、AIXでは、ブロックサイズを `lsfs -q`。Linuxの場合、`xfs_info` および `tune2fs` 次の用途に使用できます。`xfs` および `ext3/ext4` をクリックします。を使用 ``zfs`` コマンドは次のようになります。  
``zdb -C`。

ブロックサイズを制御するパラメータは次のとおりです。`ashift` 通常、デフォルト値は9です。これは  $2^9$ 、つまり512バイトを意味します。最適なパフォーマンスを実現するには、`ashift` 値は12 ( $2^{12}=4K$ ) である必要があります。この値はzpoolの作成時に設定され、変更することはできません。つまり、`ashift` 12以外の場合は、新しく作成したzpoolにデータをコピーして移行する必要があります。

Oracle ASMには基本ブロックサイズはありません。唯一の要件は、ASMディスクを構築するパーティションが適切にアライメントされていることです。

## 7-Mode Transition Tool

7-Mode Transition Tool (7MTT) は、7-Modeの大規模な構成をONTAPに移行するための自動化ユーティリティです。データベースをご利用のお客様は、ストレージの設置面積全体を移動するのではなく、データベース単位で環境のデータベースを移行することが多いため、他の方法を簡単に見つけることができます。また、多くの場合、データベースは大規模なストレージ環境の一部にすぎません。そのため、データベースは多くの場合個別に移行され、その後7MTTを使用して残りの環境を移動できます。

複雑なデータベース環境に特化したストレージシステムを運用しているお客様は少なくありませんが、かなりの数のお客様がいらっしゃいます。これらの環境には、多数のボリュームやSnapshotのほか、エクスポート権限、LUNイニシエータグループ、ユーザ権限、Lightweight Directory Access Protocolの設定など、さまざまな設定の詳細が含まれている可能性があります。このような場合は、7MTTの自動化機能によって移動が簡易化されます。

7MTTは次の2つのモードのいずれかで動作します。

- コピーベースの移行 (**CBT**)。7MTTとCBTにより、新しい環境の既存の7-ModeシステムからSnapMirrorボリュームがセットアップされます。データの同期が完了すると、7MTTによってカットオーバープロセスがオーケストレーションされます。
- コピーフリーの移行 (**CFT**)。CFTを使用する7MTTは、既存の7-Modeディスクシェルフのインプレース変換に基づいています。データはコピーされず、既存のディスクシェルフは再利用できます。データ保護とStorage Efficiencyの既存の設定は維持されます。

これら2つのオプションの主な違いは、コピーフリーの移行はビッグバンアプローチであり、元の7-Mode HAペアに接続されているすべてのディスクシェルフを新しい環境に再配置する必要がある点です。シェルフのサブセットを移動するオプションはありません。コピーベースのアプローチでは、選択したボリュームを移動できます。また、ディスクシェルフを再ケーブル接続してメタデータを変換する際にも同様の接続が必要になるため、コピーフリーの移行ではカットオーバー時間が長くなる可能性があります。NetAppでは、現場での経験に基づき、ディスクシェルフの再配置と再接続には1時間、メタデータ変換には15分から2時間かかること

を推奨しています。

## データファイルの移行

1つのコマンドで個々のOracleデータファイルを移動できます。

たとえば、次のコマンドはデータファイルIOPST.dbfをファイルシステムから移動します。 /oradata2 ファイルシステムへ /oradata3。

```
SQL> alter database move datafile  '/oradata2/NTAP/IOPS002.dbf' to
'/oradata3/NTAP/IOPS002.dbf';
Database altered.
```

この方法でデータファイルを移動すると時間がかかることがありますが、通常はI/Oが十分に発生しないため、日常のデータベースワークロードを妨げることはありません。一方、ASMのリバランシングを使用した移行ははるかに高速ですが、データの移動中にデータベース全体の処理速度が低下するという代償があります。

データファイルの移動に要する時間は、テストデータファイルを作成して移動することで簡単に測定できます。操作の経過時間は、V\$セッションデータに記録されます。

```
SQL> set linesize 300;
SQL> select elapsed_seconds||': '||message from v$session_longops;
ELAPSED_SECONDS||': '||MESSAGE
-----
-----
351:Online data file move: data file 8: 22548578304 out of 22548578304
bytes done
SQL> select bytes / 1024 / 1024 /1024 as GB from dba_data_files where
FILE_ID = 8;
          GB
-----
          21
```

この例では、移動したファイルはデータファイル8です。データファイルのサイズは21GBで、移行に約6分かかりました。必要な時間は、ストレージシステムの機能、ストレージネットワーク、および移行時に発生する全体的なデータベースアクティビティによって異なります。

## ログ配布

ログ配布を使用した移行の目的は、元のデータファイルのコピーを新しい場所に作成し、変更を新しい環境に配布する方法を確立することです。

いったん確立されると、ログの送信と再生を自動化して、レプリカデータベースをソースとほぼ同期した状態に保つことができます。たとえば、(a) 最新のログを新しい場所にコピーし、(b) 15分ごとに再生するようにcronジョブをスケジュールできます。再生が必要なアーカイブログは15分以内であるため、カットオーバー時のシステム停止は最小限に抑えられます。



次に示す手順は、基本的にはデータベースのクローニング処理です。表示されるロジックは、NetApp SnapManager for Oracle (SMO) およびNetApp SnapCenter Oracleプラグインのエンジンと似ています。一部のお客様は、スクリプトまたはWFAワークフローに表示されている手順をカスタムクローニング処理に使用しています。この手順はSMOやSnapCenterを使用するよりも手動で作成する必要がありますが、スクリプト化も容易で、ONTAP内のデータ管理APIによってプロセスがさらに簡易化されます。

## ログ配布-ファイルシステムからファイルシステムへ

この例では、Waffleというデータベースを通常のファイルシステムから別のサーバにある別の通常のファイルシステムに移行する方法を示します。また、SnapMirrorを使用してデータファイルの高速コピーを作成する方法も示していますが、これは手順全体に不可欠な要素ではありません。

### データベースバックアップの作成

まず、データベースのバックアップを作成します。具体的には、この手順には、アーカイブログの再生に使用できる一連のデータファイルが必要です。

### 環境

この例では、ソースデータベースはONTAPシステム上にあります。データベースのバックアップを作成する最も簡単な方法は、Snapshotを使用する方法です。データベースがホットバックアップモードになるまでの数秒間、`snapshot create` この処理は、データファイルをホストしているボリュームで実行されます。

```
SQL> alter database begin backup;  
Database altered.
```

```
Cluster01::*> snapshot create -vserver vserver1 -volume jfsc1_oradata  
hotbackup  
Cluster01::*>
```

```
SQL> alter database end backup;  
Database altered.
```

その結果、という名前のディスク上のスナップショットが作成されます。hotbackup ホットバックアップモード時のデータファイルのイメージを含むデータファイルを展開します。適切なアーカイブログと組み合わせでデータファイルの整合性を確保すると、このSnapshot内のデータをリストアまたはクローンのベースとして使用できます。この場合、新しいサーバに複製されます。

### 新しい環境へのリストア

これで、新しい環境でバックアップをリストアする必要があります。これは、Oracle RMAN、NetBackupなどのバックアップアプリケーションからのリストア、ホットバックアップモードに設定されたデータファイルの単純なコピー操作など、さまざまな方法で実行できます。

この例では、SnapMirrorを使用してSnapshotホットバックアップを新しい場所にレプリケートします。

1. Snapshotデータを受信する新しいボリュームを作成します。ミラーリングの初期化 jfsc1\_oradata 終

了: vol\_oradata。

```
Cluster01::*> volume create -vserver vserver1 -volume vol_oradata
-aggregate data_01 -size 20g -state online -type DP -snapshot-policy
none -policy jfsc3
[Job 833] Job succeeded: Successful
```

```
Cluster01::*> snapmirror initialize -source-path vserver1:jfsc1_oradata
-destination-path vserver1:vol_oradata
Operation is queued: snapmirror initialize of destination
"vserver1:vol_oradata".
Cluster01::*> volume mount -vserver vserver1 -volume vol_oradata
-junction-path /vol_oradata
Cluster01::*>
```

2. SnapMirrorによって同期が完了したことを示す状態が設定されたら、目的のSnapshotに基づいてミラーを更新します。

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields state
source-path          destination-path      state
-----
vserver1:jfsc1_oradata vserver1:vol_oradata SnapMirrored
```

```
Cluster01::*> snapmirror update -destination-path vserver1:vol_oradata
-source-snapshot hotbackup
Operation is queued: snapmirror update of destination
"vserver1:vol_oradata".
```

3. 同期が正常に完了したかどうかは、newest-snapshot フィールドを指定します。

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields newest-snapshot
source-path          destination-path      newest-snapshot
-----
vserver1:jfsc1_oradata vserver1:vol_oradata hotbackup
```

4. その後、ミラーを壊すことができます。

```
Cluster01::> snapmirror break -destination-path vserver1:vol_oradata
Operation succeeded: snapmirror break for destination
"vserver1:vol_oradata".
Cluster01::>
```

5. 新しいファイルシステムをマウントします。ブロックベースのファイルシステムでは、使用するLVMによって正確な手順が異なります。FCゾーニングまたはiSCSI接続を設定する必要があります。LUNへの接続が確立されたら、Linuxなどのコマンド `pvs can ASM` で検出できるように設定する必要があるボリュームグループまたはLUNを検出する場合に、が必要になることがあります。

この例では、シンプルなNFSファイルシステムを使用しています。このファイルシステムは直接マウントできます。

```
fas8060-nfs1:/vol_oradata      19922944    1639360    18283584    9%
/oradata
fas8060-nfs1:/vol_logs        9961472      128      9961344    1%
/logs
```

#### 制御ファイル作成テンプレートの作成

次に、制御ファイルテンプレートを作成する必要があります。。 `backup controlfile to trace` コマンド制御ファイルを再作成するためのテキストコマンドを作成します。この機能は、状況によってはバックアップからデータベースをリストアする場合に役立ちます。また、データベースのクローニングなどのタスクを実行するスクリプトでよく使用されます。

1. 移行されたデータベースの制御ファイルを再作成するには、次のコマンドの出力を使用します。

```
SQL> alter database backup controlfile to trace as '/tmp/waffle.ctl';
Database altered.
```

2. 制御ファイルが作成されたら、ファイルを新しいサーバにコピーします。

```
[oracle@jfsp3 tmp]$ scp oracle@jfsp1:/tmp/waffle.ctl /tmp/
oracle@jfsp1's password:
waffle.ctl                                100% 5199
5.1KB/s  00:00
```

#### バックアップパラメータファイル

新しい環境ではパラメータファイルも必要です。最も簡単な方法は、現在のspfileまたはpfileからpfileを作成することです。この例では、ソースデータベースでspfileが使用されています。

```
SQL> create pfile='/tmp/waffle.tmp.pfile' from spfile;
File created.
```

#### oratabエントリの作成

oratabエントリの作成は、oraenvなどのユーティリティが適切に機能するために必要です。oratabエントリを作成するには、次の手順を実行します。

```
WAFFLE:/orabin/product/12.1.0/dbhome_1:N
```

#### ディレクトリ構造の準備

必要なディレクトリがまだ存在していない場合は、作成する必要があります。作成しないと、データベースの起動手順が失敗します。ディレクトリ構造を準備するには、次の最小要件を満たしている必要があります。

```
[oracle@jpsc3 ~]$ . oraenv
ORACLE_SID = [oracle] ? WAFFLE
The Oracle base has been set to /orabin
[oracle@jpsc3 ~]$ cd $ORACLE_BASE
[oracle@jpsc3 orabin]$ cd admin
[oracle@jpsc3 admin]$ mkdir WAFFLE
[oracle@jpsc3 admin]$ cd WAFFLE
[oracle@jpsc3 WAFFLE]$ mkdir adump dpdump pfile scripts xdb_wallet
```

#### パラメータファイルの更新

1. パラメータファイルを新しいサーバにコピーするには、次のコマンドを実行します。デフォルトの場所は \$ORACLE\_HOME/dbs ディレクトリ。この場合、pfileは任意の場所に配置できます。これは、移行プロセスの中間ステップとしてのみ使用されます。

```
[oracle@jpsc3 admin]$ scp oracle@jpsc1:/tmp/waffle.tmp.pfile
$ORACLE_HOME/dbs/waffle.tmp.pfile
oracle@jpsc1's password:
waffle.pfile                                100%  916
0.9KB/s   00:00
```

1. 必要に応じてファイルを編集します。たとえば、アーカイブログの場所が変更された場合は、新しい場所を反映するようにpfileを変更する必要があります。この例では、制御ファイルだけが再配置されています。その一部は、ログファイルシステムとデータファイルシステム間で制御ファイルを分散するためです。

```
[root@jfscl tmp]# cat waffle.pfile
WAFFLE.__data_transfer_cache_size=0
WAFFLE.__db_cache_size=507510784
WAFFLE.__java_pool_size=4194304
WAFFLE.__large_pool_size=20971520
WAFFLE.__oracle_base='/orabin'#ORACLE_BASE set from environment
WAFFLE.__pga_aggregate_target=268435456
WAFFLE.__sga_target=805306368
WAFFLE.__shared_io_pool_size=29360128
WAFFLE.__shared_pool_size=234881024
WAFFLE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/WAFFLE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/oradata//WAFFLE/control01.ctl','/oradata//WAFFLE/control02.ctl'
*.control_files='/oradata/WAFFLE/control01.ctl','/logs/WAFFLE/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='WAFFLE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=WAFFLEXDB)'
*.log_archive_dest_1='LOCATION=/logs/WAFFLE/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

2. 編集が完了したら、このpfileに基づいてspfileを作成します。

```
SQL> create spfile from pfile='waffle.tmp.pfile';
File created.
```

#### 制御ファイルの再作成

前の手順では、`backup controlfile to trace` が新しいサーバにコピーされました。必要な出力の具体的な部分は、`controlfile recreation` コマンドを実行しますこの情報は、ファイルのマークされたセクションの下に記載されています。Set #1. NORESETLOGS。次の行から始まります `create controlfile reuse database` 次の単語を含める必要があります。 `noresetlogs`。最後はセミコロン (;) 文字です。

1. この手順の例では、ファイルは次のように表示されます。

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/logs/WAFFLE/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/logs/WAFFLE/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/logs/WAFFLE/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

2. このスクリプトを必要に応じて編集し、さまざまなファイルの新しい場所を反映します。たとえば、高I/Oをサポートすると認識されている特定のデータファイルは、ハイパフォーマンスストレージ階層上のファイルシステムにリダイレクトされる可能性があります。また、特定のPDBのデータファイルを専用ボリュームに分離するなど、管理者のみが変更を行う場合もあります。
3. この例では、を使用しています DATAFILE スタンザは変更されませんが、REDOログは /redo アーカイブログでスペースを共有する代わりに /logs。



```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

```

SQL> startup nomount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size              331353200 bytes
Database Buffers           465567744 bytes
Redo Buffers                5455872 bytes
SQL> CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
 2      MAXLOGFILES 16
 3      MAXLOGMEMBERS 3
 4      MAXDATAFILES 100
 5      MAXINSTANCES 8
 6      MAXLOGHISTORY 292
 7  LOGFILE
 8      GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
 9      GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
10      GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
11  -- STANDBY LOGFILE
12  DATAFILE
13      '/oradata/WAFFLE/system01.dbf',
14      '/oradata/WAFFLE/sysaux01.dbf',
15      '/oradata/WAFFLE/undotbs01.dbf',
16      '/oradata/WAFFLE/users01.dbf'
17  CHARACTER SET WE8MSWIN1252
18  ;
Control file created.
SQL>

```

ファイルが正しく配置されていない場合やパラメータが正しく設定されていない場合は、修正が必要な項目を示すエラーが生成されます。データベースはマウントされていますが、使用中のデータファイルがホットバックアップモードとしてマークされているため、まだ開いておらず、開くことができません。データベースの整合性を維持するには、まずアーカイブログを適用する必要があります。

#### 初期ログレプリケーション

データファイルの整合性を確保するには、少なくとも1つのログ応答処理が必要です。ログの再生には、さまざまなオプションを使用できます。場合によっては、元のサーバ上の元のアーカイブログの場所をNFS経由で共有し、ログの返信を直接行うことができます。それ以外の場合は、アーカイブログをコピーする必要があります。

例えば、単純な scp この処理では、現在のすべてのログを移行元サーバから移行先サーバにコピーできます。

```

[oracle@jpsc3 arch]$ scp jpsc1:/logs/WAFFLE/arch/* ./
oracle@jpsc1's password:
1_22_912662036.dbf                                100%   47MB
47.0MB/s   00:01
1_23_912662036.dbf                                100%   40MB
40.4MB/s   00:00
1_24_912662036.dbf                                100%   45MB
45.4MB/s   00:00
1_25_912662036.dbf                                100%   41MB
40.9MB/s   00:01
1_26_912662036.dbf                                100%   39MB
39.4MB/s   00:00
1_27_912662036.dbf                                100%   39MB
38.7MB/s   00:00
1_28_912662036.dbf                                100%   40MB
40.1MB/s   00:01
1_29_912662036.dbf                                100%   17MB
16.9MB/s   00:00
1_30_912662036.dbf                                100%   636KB
636.0KB/s   00:00

```

#### 初回のログ再生

アーカイブログの場所に保存されたファイルは、コマンドを実行して再生できます。recover database until cancel その後に応答が続きます AUTO 使用可能なすべてのログを自動的に再生します。

```

SQL> recover database until cancel;
ORA-00279: change 382713 generated at 05/24/2016 09:00:54 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_23_912662036.dbf
ORA-00280: change 382713 for thread 1 is in sequence #23
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 405712 generated at 05/24/2016 15:01:05 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_24_912662036.dbf
ORA-00280: change 405712 for thread 1 is in sequence #24
ORA-00278: log file '/logs/WAFFLE/arch/1_23_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
ORA-00278: log file '/logs/WAFFLE/arch/1_30_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_31_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

```

最後のアーカイブログの応答でエラーが報告されますが、これは正常な動作です。ログは次のことを示します。sqlplus 特定のログファイルを探していましたが、見つかりませんでした。ログファイルがまだ存在しない可能性があります。

アーカイブログをコピーする前にソースデータベースをシャットダウンできる場合、この手順は1回だけ実行する必要があります。アーカイブログがコピーされて再生されたら、重要なRedoログをレプリケートするカットオーバープロセスに直接進むことができます。

#### 差分ログのレプリケーションと再生

ほとんどの場合、移行はすぐには実行されません。移行プロセスが完了するまでに数日、場合によっては数週間かかることもあります。つまり、ログをレプリカデータベースに継続的に送信して再生する必要があります。そのため、カットオーバーが完了したら、最小限のデータを転送して再生する必要があります。

これはさまざまな方法でスクリプト化できますが、最も一般的な方法の1つは、一般的なファイルレプリケーションユーティリティであるrsyncを使用することです。このユーティリティを使用する最も安全な方法は、このユーティリティをデーモンとして設定することです。たとえば、などです rsyncd.conf 次のファイルは、という名前のリソースを作成する方法を示しています。 waffle.arch Oracleユーザクレデンシャルでアクセスされ、次にマッピングされます。 /logs/WAFFLE/arch。最も重要なことは、リソースが読み取り専用を設定されていることです。これにより、本番データの読み取りは可能ですが、変更はできません。

```
[root@jfscl arch]# cat /etc/rsyncd.conf
[waffle.arch]
uid=oracle
gid=dba
path=/logs/WAFFLE/arch
read only = true
[root@jfscl arch]# rsync --daemon
```

次のコマンドは新しいサーバのアーカイブログデスティネーションをrsyncリソースと同期します  
waffle.arch 元のサーバ。。 t の引数 rsync - potg タイムスタンプに基づいてファイルリストが比較され、新しいファイルのみがコピーされます。このプロセスでは、新しいサーバの増分アップデートが提供されます。このコマンドは、cronで定期的に行うようにスケジュールすることもできます。

```

[oracle@jfsc3 arch]$ rsync -potg --stats --progress jfsc1::waffle.arch/*
/logs/WAFFLE/arch/
1_31_912662036.dbf
    650240 100% 124.02MB/s    0:00:00 (xfer#1, to-check=8/18)
1_32_912662036.dbf
    4873728 100% 110.67MB/s    0:00:00 (xfer#2, to-check=7/18)
1_33_912662036.dbf
    4088832 100%  50.64MB/s    0:00:00 (xfer#3, to-check=6/18)
1_34_912662036.dbf
    8196096 100%  54.66MB/s    0:00:00 (xfer#4, to-check=5/18)
1_35_912662036.dbf
    19376128 100%  57.75MB/s    0:00:00 (xfer#5, to-check=4/18)
1_36_912662036.dbf
     71680 100% 201.15kB/s    0:00:00 (xfer#6, to-check=3/18)
1_37_912662036.dbf
    1144320 100%   3.06MB/s    0:00:00 (xfer#7, to-check=2/18)
1_38_912662036.dbf
    35757568 100%  63.74MB/s    0:00:00 (xfer#8, to-check=1/18)
1_39_912662036.dbf
     984576 100%   1.63MB/s    0:00:00 (xfer#9, to-check=0/18)
Number of files: 18
Number of files transferred: 9
Total file size: 399653376 bytes
Total transferred file size: 75143168 bytes
Literal data: 75143168 bytes
Matched data: 0 bytes
File list size: 474
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 204
Total bytes received: 75153219
sent 204 bytes  received 75153219 bytes  150306846.00 bytes/sec
total size is 399653376  speedup is 5.32

```

ログを受信したら、それらのログを再生する必要があります。上記の例では、sqlplusを使用して手動で recover database until cancel、簡単に自動化できるプロセス。この例では、で説明されているスクリプトを使用しています。"データベースのログを再生"。スクリプトは、リプレイ操作を必要とするデータベースを指定する引数を受け入れます。これにより、同じスクリプトをマルチデータベース移行で使用できます。



```

[oracle@jpsc3 logs]$ ./replay.logs.pl WAFFLE
ORACLE_SID = [WAFFLE] ? The Oracle base remains unchanged with value
/orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu May 26 10:47:16 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 814256 generated at 05/26/2016 04:52:30 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_32_912662036.dbf
ORA-00280: change 814256 for thread 1 is in sequence #32
ORA-00278: log file '/logs/WAFFLE/arch/1_31_912662036.dbf' no longer
needed for
this recovery
ORA-00279: change 814780 generated at 05/26/2016 04:53:04 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_33_912662036.dbf
ORA-00280: change 814780 for thread 1 is in sequence #33
ORA-00278: log file '/logs/WAFFLE/arch/1_32_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
ORA-00278: log file '/logs/WAFFLE/arch/1_39_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options

```

新しい環境にカットオーバーする準備ができれば、アーカイブログとREDOログの両方を含む最終的な同期を実行する必要があります。元のREDOログの場所が不明な場合は、次のように特定できます。

```
SQL> select member from v$logfile;
MEMBER
-----
-----
/logs/WAFFLE/redo/redo01.log
/logs/WAFFLE/redo/redo02.log
/logs/WAFFLE/redo/redo03.log
```

1. ソースデータベースをシャットダウンします。
2. 目的の方法を使用して、新しいサーバでアーカイブログの最終的な同期を1回実行します。
3. ソースREDOログを新しいサーバにコピーする必要があります。この例では、REDOログがの新しいディレクトリに再配置されています。 /redo。

```
[oracle@jfspc3 logs]$ scp jfspc1:/logs/WAFFLE/redo/* /redo/
oracle@jfspc1's password:
redo01.log
100% 50MB 50.0MB/s 00:01
redo02.log
100% 50MB 50.0MB/s 00:00
redo03.log
100% 50MB 50.0MB/s 00:00
```

4. この段階で、新しいデータベース環境には、ソースとまったく同じ状態にするために必要なすべてのファイルが含まれています。アーカイブログは最後に1回再生する必要があります。

```

SQL> recover database until cancel;
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

```

5. 完了したら、Redoログを再生する必要があります。というメッセージが表示されます Media recovery complete が返されると、プロセスが成功し、データベースが同期されてオープンできるようになります。

```

SQL> recover database;
Media recovery complete.
SQL> alter database open;
Database altered.

```

## ログ配布-ASMからファイルシステムへ

この例では、Oracle RMANを使用してデータベースを移行します。ファイルシステムからファイルシステムへのログ配布の前の例と非常によく似ていますが、ASM上のファイルはホストには表示されません。ASMデバイス上にあるデータを移行するには、ASM LUNを再配置するか、Oracle RMANを使用してコピー処理を実行するしかありません。

Oracle ASMからファイルをコピーするにはRMANが必要ですが、RMANを使用できるのはASMに限られません。RMANを使用すると、任意のタイプのストレージから他のタイプのストレージに移行できます。

この例は'pancake'というデータベースをASMストレージから'パス'にある別のサーバにある通常のファイルシステムに再配置する例を示しています /oradata および /logs。

### データベースバックアップの作成

最初の手順では、代替サーバに移行するデータベースのバックアップを作成します。ソースではOracle ASMを使用するため、RMANを使用する必要があります。単純なRMANバックアップは、次のように実行できます。この方法で作成されるタグ付きバックアップは、あとでRMANで簡単に識別できるように手順になります。

最初のコマンドは、バックアップ先のタイプと使用する場所を定義します。2番目のコマンドでは、データファイルのみのバックアップが開始されます。

```
RMAN> configure channel device type disk format '/rman/pancake/%U';
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT    '/rman/pancake/%U';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT    '/rman/pancake/%U';
new RMAN configuration parameters are successfully stored
RMAN> backup database tag 'ONTAP_MIGRATION';
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=251 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/PANCAKE/system01.dbf
input datafile file number=00002 name=+ASM0/PANCAKE/sysaux01.dbf
input datafile file number=00003 name=+ASM0/PANCAKE/undotbs101.dbf
input datafile file number=00004 name=+ASM0/PANCAKE/users01.dbf
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/lgr6c161_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/lhr6c164_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

バックアップ制御ファイルバックアップセイギョファイル

バックアップ制御ファイルは、手順の後半の工程で duplicate database 操作。

```
RMAN> backup current controlfile format '/rman/pancake/ctrl.bkp';
Starting backup at 24-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/ctrl.bkp tag=TAG20160524T032651 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

#### バックアップパラメータファイル

新しい環境ではパラメータファイルも必要です。最も簡単な方法は、現在のspfileまたはpfileからpfileを作成することです。この例では、ソースデータベースでspfileが使用されています。

```
RMAN> create pfile='/rman/pancake/pfile' from spfile;
Statement processed
```

#### ASMファイル名変更スクリプト

データベースを移動すると、制御ファイルに現在定義されている複数のファイルの場所が変更されます。次のスクリプトは、プロセスを簡単にするためにRMANスクリプトを作成します。この例は、データファイルの数が非常に少ないデータベースを示していますが、通常、データベースには数百、場合によっては数千のデータファイルが含まれています。

このスクリプトは、["ASMからファイルシステム名への変換"](#) 2つのことができます

まず、REDOログの場所を再定義するパラメータを作成します。log\_file\_name\_convert。基本的には交互のフィールドのリストです。最初のフィールドは現在のREDOログの場所で、2番目のフィールドは新しいサーバ上の場所です。その後、パターンが繰り返されます。

2つ目の機能は、データファイルの名前を変更するためのテンプレートを提供することです。スクリプトは、データファイルをループ処理し、名前とファイル番号の情報を取得して、RMANスクリプトとしてフォーマットします。次に、一時ファイルについても同じことが行われます。その結果、必要に応じて編集してファイルを目的の場所にリストアできるシンプルなRMANスクリプトが作成されます。

```

SQL> @/rman/mk.rename.scripts.sql
Parameters for log file conversion:
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/NEW_PATH/redo01.log', '+ASM0/PANCAKE/redo02.log',
'/NEW_PATH/redo02.log', '+ASM0/PANCAKE/redo03.log', '/NEW_PATH/redo03.log'
rman duplication script:
run
{
set newname for datafile 1 to '+ASM0/PANCAKE/system01.dbf';
set newname for datafile 2 to '+ASM0/PANCAKE/sysaux01.dbf';
set newname for datafile 3 to '+ASM0/PANCAKE/undotbs101.dbf';
set newname for datafile 4 to '+ASM0/PANCAKE/users01.dbf';
set newname for tempfile 1 to '+ASM0/PANCAKE/temp01.dbf';
duplicate target database for standby backup location INSERT_PATH_HERE;
}
PL/SQL procedure successfully completed.

```

この画面の出力をキャプチャします。。log\_file\_name\_convert パラメータは、次のようにpfileに配置されます。RMANデータ・ファイルの名前変更および複製スクリプトを編集して、必要な場所にデータ・ファイルを配置する必要があります。この例では、これらはすべて /oradata/pancake。

```

run
{
set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
set newname for datafile 4 to '/oradata/pancake/users.dbf';
set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
duplicate target database for standby backup location '/rman/pancake';
}

```

#### ディレクトリ構造の準備

スクリプトの実行準備はほぼ完了していますが、最初にディレクトリ構造を設定する必要があります。必要なディレクトリが存在しない場合は、それらのディレクトリを作成する必要があります。存在しないと、データベースの起動手順が失敗します。次の例は、最小要件を示しています。

```

[oracle@jfspc2 ~]$ mkdir /oradata/pancake
[oracle@jfspc2 ~]$ mkdir /logs/pancake
[oracle@jfspc2 ~]$ cd /orabin/admin
[oracle@jfspc2 admin]$ mkdir PANCAKE
[oracle@jfspc2 admin]$ cd PANCAKE
[oracle@jfspc2 PANCAKE]$ mkdir adump dpdump pfile scripts xdb_wallet

```



## oratabエントリの作成

次のコマンドは、oraenvなどのユーティリティが正常に動作するために必要です。

```
PANCAKE:/orabin/product/12.1.0/dbhome_1:N
```

## パラメータの更新

保存したpfileを更新して、新しいサーバ上のパスの変更を反映する必要があります。データ・ファイル・パスの変更は、RMAN複製スクリプトによって変更されます。ほとんどのデータベースでは、`control_files` および `log_archive_dest` パラメータ変更が必要な監査ファイルの場所や、次のようなパラメータが存在する場合もあります。`db_create_file_dest` ASM以外では関連性がない可能性があります。経験豊富なデータベース管理者は、次に進む前に提案された変更を慎重に確認する必要があります。

この例では、制御ファイルの場所、ログのアーカイブ先、`log_file_name_convert` パラメータ

```

PANCAKE.__data_transfer_cache_size=0
PANCAKE.__db_cache_size=545259520
PANCAKE.__java_pool_size=4194304
PANCAKE.__large_pool_size=25165824
PANCAKE.__oracle_base='/orabin'#ORACLE_BASE set from environment
PANCAKE.__pga_aggregate_target=268435456
PANCAKE.__sga_target=805306368
PANCAKE.__shared_io_pool_size=29360128
PANCAKE.__shared_pool_size=192937984
PANCAKE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/PANCAKE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='+ASM0/PANCAKE/control01.ctl','+ASM0/PANCAKE/control02.ctl'
*.control_files='/oradata/pancake/control01.ctl','/logs/pancake/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='PANCAKE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=PANCAKEXDB)'
*.log_archive_dest_1='LOCATION=+ASM1'
*.log_archive_dest_1='LOCATION=/logs/pancake'
*.log_archive_format='%t_%s_%r.dbf'
'/logs/path/redo02.log'
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/logs/pancake/redo01.log', '+ASM0/PANCAKE/redo02.log',
'/logs/pancake/redo02.log', '+ASM0/PANCAKE/redo03.log',
'/logs/pancake/redo03.log'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'

```

新しいパラメータが確認されたら、パラメータを有効にする必要があります。複数のオプションがありますが、ほとんどのお客様はテキストfileに基づいてspfileを作成します。

```
bash-4.1$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Fri Jan 8 11:17:40 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> create spfile from pfile='/rman/pancake/pfile';
File created.
```

#### スタートアップの登録

データベースをレプリケートする前の最後の手順では、データベースプロセスを起動しますが、ファイルはマウントしません。この手順では、spfileの問題が明らかになる可能性があります。状況に応じて startup nomount パラメータエラーが原因でコマンドが失敗します。シャットダウンし、pfileテンプレートを修正し、spfileとしてリロードして、再試行するのは簡単です。

```
SQL> startup nomount;
ORACLE instance started.
Total System Global Area 805306368 bytes
Fixed Size 2929552 bytes
Variable Size 373296240 bytes
Database Buffers 423624704 bytes
Redo Buffers 5455872 bytes
```

#### データベースの複製

以前のRMANバックアップを新しい場所にリストアするには、このプロセスの他の手順よりも時間がかかります。データベースID (DBID) を変更したり、ログをリセットしたりせずに、データベースを複製する必要があります。これにより、ログが適用されなくなります。これは、コピーを完全に同期するために必要な手順です。

前の手順で作成したスクリプトを使用して、RMANをauxとしてデータベースに接続し、DUPLICATE DATABASEコマンドを問題します。

```
[oracle@jfsc2 pancake]$ rman auxiliary /
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 03:04:56
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights
reserved.
connected to auxiliary database: PANCAKE (not mounted)
RMAN> run
2> {
3> set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
4> set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
5> set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
6> set newname for datafile 4 to '/oradata/pancake/users.dbf';
7> set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
```

```

8> duplicate target database for standby backup location '/rman/pancake';
9> }
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting Duplicate Db at 24-MAY-16
contents of Memory Script:
{
    restore clone standby controlfile from  '/rman/pancake/ctrl.bkp';
}
executing Memory Script
Starting restore at 24-MAY-16
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file name=/oradata/pancake/control01.ctl
output file name=/logs/pancake/control02.ctl
Finished restore at 24-MAY-16
contents of Memory Script:
{
    sql clone 'alter database mount standby database';
}
executing Memory Script
sql statement: alter database mount standby database
released channel: ORA_AUX_DISK_1
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
contents of Memory Script:
{
    set newname for tempfile 1 to
    "/oradata/pancake/temp.dbf";
    switch clone tempfile all;
    set newname for datafile 1 to
    "/oradata/pancake/pancake.dbf";
    set newname for datafile 2 to
    "/oradata/pancake/sysaux.dbf";
    set newname for datafile 3 to
    "/oradata/pancake/undotbs1.dbf";
    set newname for datafile 4 to
    "/oradata/pancake/users.dbf";
    restore
    clone database
    ;
}

```

```

}
executing Memory Script
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/pancake/temp.dbf in control file
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting restore at 24-MAY-16
using channel ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/oradata/pancake/pancake.dbf
channel ORA_AUX_DISK_1: restoring datafile 00002 to
/oradata/pancake/sysaux.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/oradata/pancake/undotbs1.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/oradata/pancake/users.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/rman/pancake/1gr6c161_1_1
channel ORA_AUX_DISK_1: piece handle=/rman/pancake/1gr6c161_1_1
tag=ONTAP_MIGRATION
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:07
Finished restore at 24-MAY-16
contents of Memory Script:
{
    switch clone datafile all;
}
executing Memory Script
datafile 1 switched to datafile copy
input datafile copy RECID=5 STAMP=912655725 file
name=/oradata/pancake/pancake.dbf
datafile 2 switched to datafile copy
input datafile copy RECID=6 STAMP=912655725 file
name=/oradata/pancake/sysaux.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=7 STAMP=912655725 file
name=/oradata/pancake/undotbs1.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=8 STAMP=912655725 file
name=/oradata/pancake/users.dbf
Finished Duplicate Db at 24-MAY-16

```

ソースデータベースから新しい場所に変更を出荷する必要があります。そのためには、いくつかの手順が必要になる場合があります。最も簡単な方法は、ソース・データベースのRMANでアーカイブ・ログを共有ネットワーク接続に書き込む方法です。共有の場所を使用できない場合は、RMANを使用してローカルファイルシステムに書き込み、`rcp`または`rsync`を使用してファイルをコピーする方法もあります。

この例では、を使用しています `/rman` ディレクトリは、元のデータベースと移行後のデータベースの両方で使用できるNFS共有です。

ここでの重要な問題の1つは、`disk format` 条項。バックアップのディスクフォーマットは次のとおりです。 `%h_%e_%a.dbf` これは、スレッド番号、シーケンス番号、およびデータベースのアクティベーションIDの形式を使用する必要があることを意味します。文字は異なりますが、これは ``log_archive_format='%t_%s_%r.dbf` パラメータを`pfile`に指定します。このパラメータは、スレッド番号、シーケンス番号、およびアクティベーションIDの形式でアーカイブログを指定します。最終的に、ソース上のログファイルのバックアップでは、データベースで想定される命名規則が使用されます。これにより、次のような操作が行われます。 `recover database sqlplus`はアーカイブログの名前を正しく予測して再生できるため、はるかにシンプルです。

```

RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/arch/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
released channel: ORA_DISK_1
RMAN> backup as copy archivelog from time 'sysdate-2';
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=373 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=70 STAMP=912658508
output file name=/rman/pancake/logship/1_54_912576125.dbf RECID=123
STAMP=912659482
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=29 STAMP=912654101
output file name=/rman/pancake/logship/1_41_912576125.dbf RECID=124
STAMP=912659483
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=33 STAMP=912654688
output file name=/rman/pancake/logship/1_45_912576125.dbf RECID=152
STAMP=912659514
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=36 STAMP=912654809
output file name=/rman/pancake/logship/1_47_912576125.dbf RECID=153
STAMP=912659515
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16

```

#### 初回のログ再生

アーカイブログの場所に保存されたファイルは、コマンドを実行して再生できます。recover database until cancel その後に応答が続きます AUTO 使用可能なすべてのログを自動的に再生します。パラメータファイルは現在、アーカイブログを次の場所に転送しています：`/logs/archive`ただし、これは、RMANを使用してログを保存した場所と一致しません。この場所は、データベースをリカバリする前に、次のように一時的にリダイレクトできます。

```

SQL> alter system set log_archive_dest_1='LOCATION=/rman/pancake/logship'
scope=memory;
System altered.
SQL> recover standby database until cancel;
ORA-00279: change 560224 generated at 05/24/2016 03:25:53 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_49_912576125.dbf
ORA-00280: change 560224 for thread 1 is in sequence #49
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 560353 generated at 05/24/2016 03:29:17 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_50_912576125.dbf
ORA-00280: change 560353 for thread 1 is in sequence #50
ORA-00278: log file '/rman/pancake/logship/1_49_912576125.dbf' no longer
needed
for this recovery
...
ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
ORA-00278: log file '/rman/pancake/logship/1_53_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_54_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

```

最後のアーカイブログの応答でエラーが報告されますが、これは正常な動作です。エラーは、sqlplusが特定のログファイルを探していたが見つからなかったことを示しています。ログファイルがまだ存在しない可能性があります。

アーカイブログをコピーする前にソースデータベースをシャットダウンできる場合、この手順は1回だけ実行する必要があります。アーカイブログがコピーされて再生されたら、重要なRedoログをレプリケートするカットオーバープロセスに直接進むことができます。

#### 差分ログのレプリケーションと再生

ほとんどの場合、移行はすぐには実行されません。移行プロセスが完了するまでに数日、場合によっては数週間かかることもあります。つまり、ログをレプリカデータベースに継続的に送信して再生する必要があります。これにより、カットオーバーの到着時に最小限のデータの転送と再生が必要になります。

このプロセスは簡単にスクリプト化できます。たとえば、次のコマンドを元のデータベースでスケジュールして、ログ配布に使用される場所が継続的に更新されるようにすることができます。



```
[oracle@jfscl pancake]$ cat copylogs.rman
configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
backup as copy archivelog from time 'sysdate-2';
```

```
[oracle@jfscl pancake]$ rman target / cmdfile=copylogs.rman
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 04:36:19
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights
reserved.
connected to target database: PANCAKE (DBID=3574534589)
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=369 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:22
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=124 STAMP=912659483
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:23
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_41_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
```

```
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:55
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:57
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf
Recovery Manager complete.
```

ログを受信したら、それらのログを再生する必要があります。上記の例では、sqlplusを使用して手動で`recover database until cancel`をクリックします。これは簡単に自動化できます。この例では、で説明されているスクリプトを使用しています。"[スタンバイデータベースのリプレイログ](#)"。スクリプトは、リプレイ操作を必要とするデータベースを指定する引数を受け取ります。このプロセスでは、同じスクリプトをマルチデータベース移行で使用できます。

```

[root@jffsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 04:47:10 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 562219 generated at 05/24/2016 04:15:08 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_55_912576125.dbf
ORA-00280: change 562219 for thread 1 is in sequence #55
ORA-00278: log file '/rman/pancake/logship/1_54_912576125.dbf' no longer
needed for this recovery
ORA-00279: change 562370 generated at 05/24/2016 04:19:18 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_56_912576125.dbf
ORA-00280: change 562370 for thread 1 is in sequence #56
ORA-00278: log file '/rman/pancake/logship/1_55_912576125.dbf' no longer
needed for this recovery
...
ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
ORA-00278: log file '/rman/pancake/logship/1_64_912576125.dbf' no longer
needed for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_65_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options

```

新しい環境にカットオーバーする準備ができれば、最後の同期を1回実行する必要があります。通常のファイルシステムを使用する場合は、元のREDOログがコピーされて再生されるため、移行したデータベースが元のデータベースと完全に同期されていることを簡単に確認できます。ASMでこれを行う良い方法はありません。簡単に再コピーできるのはアーカイブログだけです。データが失われないようにするには、元のデータベースの最終的なシャットダウンを慎重に実行する必要があります。

1. まず、データベースを休止して、変更が行われていないことを確認する必要があります。この休止には、スケジュールされた処理の無効化、リスナーのシャットダウン、アプリケーションのシャットダウンなどが含まれます。
2. この手順を実行すると、ほとんどのDBAはダミーテーブルを作成し、シャットダウンのマーカーとして機能します。
3. ログを強制的にアーカイブし、ダミーテーブルの作成がアーカイブログに記録されるようにします。これを行うには、次のコマンドを実行します。

```
SQL> create table cutovercheck as select * from dba_users;
Table created.
SQL> alter system archive log current;
System altered.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

4. 最後のアーカイブログをコピーするには、次のコマンドを実行します。データベースは使用可能であるが、開いていない必要があります。

```
SQL> startup mount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size              331353200 bytes
Database Buffers           465567744 bytes
Redo Buffers                5455872 bytes
Database mounted.
```

5. アーカイブログをコピーするには、次のコマンドを実行します。

```

RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=8 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:24
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:58
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:59:00
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf

```

6. 最後に、残りのアーカイブログを新しいサーバで再生します。

```

[root@jpsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 05:00:53 2016
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed
for thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 563629 generated at 05/24/2016 04:55:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_66_912576125.dbf
ORA-00280: change 563629 for thread 1 is in sequence #66
ORA-00278: log file '/rman/pancake/logship/1_65_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_66_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options

```

7. この段階では、すべてのデータをレプリケートします。データベースをスタンバイデータベースからアクティブ運用データベースに変換してオープンする準備が整いました。

```

SQL> alter database activate standby database;
Database altered.
SQL> alter database open;
Database altered.

```

8. ダミーテーブルの存在を確認してからドロップします。

```

SQL> desc cutovercheck
      Name                                         Null?      Type
-----
-----
      USERNAME                                   NOT NULL   VARCHAR2(128)
      USER_ID                                    NOT NULL   NUMBER
      PASSWORD                                            VARCHAR2(4000)
      ACCOUNT_STATUS                             NOT NULL   VARCHAR2(32)
      LOCK_DATE                                            DATE
      EXPIRY_DATE                                         DATE
      DEFAULT_TABLESPACE                         NOT NULL   VARCHAR2(30)
      TEMPORARY_TABLESPACE                       NOT NULL   VARCHAR2(30)
      CREATED                                    NOT NULL   DATE
      PROFILE                                    NOT NULL   VARCHAR2(128)
      INITIAL_RSRC_CONSUMER_GROUP                         VARCHAR2(128)
      EXTERNAL_NAME                                       VARCHAR2(4000)
      PASSWORD_VERSIONS                                   VARCHAR2(12)
      EDITIONS_ENABLED                                    VARCHAR2(1)
      AUTHENTICATION_TYPE                                 VARCHAR2(8)
      PROXY_ONLY_CONNECT                                  VARCHAR2(1)
      COMMON                                               VARCHAR2(3)
      LAST_LOGIN                                          TIMESTAMP(9) WITH
TIME  ZONE
      ORACLE_MAINTAINED                                   VARCHAR2(1)
SQL> drop table cutovercheck;
Table dropped.

```

## Redoログの無停止移行

REDOログを除き、データベース全体が正しく構成されている場合があります。これはさまざまな理由で発生する可能性があります。最も一般的なのはスナップショットに関連しています。SnapManager for Oracle、SnapCenter、NetApp Snap Creatorのストレージ管理フレームワークなどの製品では、データファイルボリュームの状態をリバートする場合にのみ、データベースをほぼ瞬時にリカバリできます。REDOログがデータファイルとスペースを共有している場合は、REDOログが破棄されてデータが失われる可能性があるため、リバートを安全に実行できません。そのため、REDOログを再配置する必要があります。

この手順はシンプルで、無停止で実行できます。

### 現在のREDOログ設定

1. REDOロググループの数とそれぞれのグループ番号を確認します。

```
SQL> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
rows selected.
```

## 2. Redoログのサイズを入力します。

```
SQL> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 524288000
2 524288000
3 524288000
```

### 新しいログを作成する

#### 1. Redoログごとに、サイズとメンバー数が一致する新しいグループを作成します。

```
SQL> alter database add logfile ('/newredo0/redo01a.log',
'/newredo1/redo01b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo02a.log',
'/newredo1/redo02b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo03a.log',
'/newredo1/redo03b.log') size 500M;
Database altered.
SQL>
```

#### 2. 新しい設定を確認します。



```
SQL> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
4 /newredo0/redo01a.log
4 /newredo1/redo01b.log
5 /newredo0/redo02a.log
5 /newredo1/redo02b.log
6 /newredo0/redo03a.log
6 /newredo1/redo03b.log
12 rows selected.
```

#### 古いログを削除

1. 古いログ（グループ1、2、3）を削除します。

```
SQL> alter database drop logfile group 1;
Database altered.
SQL> alter database drop logfile group 2;
Database altered.
SQL> alter database drop logfile group 3;
Database altered.
```

2. アクティブなログをドロップできないエラーが発生した場合は、次のログに切り替えてロックを解除し、グローバルチェックポイントを強制的に実行します。このプロセスの次の例を参照してください。古い場所にあるログファイルグループ2を削除しようとしたが、このログファイルにアクティブなデータが残っているため拒否されました。

```
SQL> alter database drop logfile group 2;
alter database drop logfile group 2
*
ERROR at line 1:
ORA-01623: log 2 is current log for instance NTAP (thread 1) - cannot
drop
ORA-00312: online log 2 thread 1: '/redo0/NTAP/redo02a.log'
ORA-00312: online log 2 thread 1: '/redo1/NTAP/redo02b.log'
```

3. ログアーカイブの後にチェックポイントを追加すると、ログファイルをドロップできます。

```
SQL> alter system archive log current;
System altered.
SQL> alter system checkpoint;
System altered.
SQL> alter database drop logfile group 2;
Database altered.
```

4. 次に、ファイルシステムからログを削除します。このプロセスは細心の注意を払って実行する必要があります。

## ホストデータのコピー

データベースレベルの移行と同様に、ホストレイヤでの移行では、ストレージベンダーに依存しないアプローチが提供されます。

言い換えれば、いつか「ファイルをコピーするだけ」が最良のオプションです。

このローテクなアプローチは基本的すぎるように思われるかもしれませんが、特別なソフトウェアは必要なく、プロセス中に元のデータに安全に触れることができないため、大きな利点があります。主な制限事項は、ファイルコピーデータの移行はシステムの停止を伴うプロセスであることです。これは、コピー処理を開始する前にデータベースをシャットダウンする必要があるためです。ファイル内の変更を同期する適切な方法はなため、コピーを開始する前にファイルを完全に休止する必要があります。

コピー処理で必要なシャットダウンが望ましくない場合、次に推奨されるホストベースのオプションは論理ボリュームマネージャ（LVM）を利用することです。Oracle ASMを含む多くのLVMオプションは、すべて同様の機能を備えていますが、いくつかの制限事項を考慮する必要があります。ほとんどの場合、移行はダウンタイムやシステム停止なしで完了します。

## ファイルシステムからファイルシステムへのコピー

単純なコピー操作の有用性を過小評価してはなりません。この処理はコピープロセス中のダウンタイムを必要としますが、信頼性の高いプロセスであり、オペレーティングシステム、データベース、ストレージシステムに関する特別な専門知識は必要ありません。さらに、元のデータに影響を与えないため、非常に安全です。通常システム管理者は'ソース・ファイル・システムを読み取り専用としてマウントするように変更してから'サーバを再起動して'現在のデータに損傷を与えないようにします'コピープロセスをスクリプト化して、ユーザーエラーのリスクなしにできるだけ迅速に実行できるようにすることができます。I/Oのタイプはデータの単純なシーケンシャル転送であるため、帯域幅効率に優れています。

次の例は、安全かつ迅速な移行のための1つのオプションを示しています。

環境

移行する環境は次のとおりです。

- 現在のファイルシステム

```

ontap-nfs1:/host1_oradata      52428800  16196928  36231872  31%
/oradata
ontap-nfs1:/host1_logs        49807360   548032  49259328   2% /logs

```

#### • 新しいファイルシステム

```

ontap-nfs1:/host1_logs_new      49807360      128  49807232   1%
/new/logs
ontap-nfs1:/host1_oradata_new  49807360      128  49807232   1%
/new/oradata

```

## 概要

データベースは、データベースをシャットダウンしてファイルをコピーするだけで移行できますが、多数のデータベースを移行する必要がある場合や、ダウンタイムを最小限に抑えることが重要な場合は、プロセスを簡単にスクリプト化できます。スクリプトを使用すると、ユーザエラーの可能性も低くなります。

このスクリプトの例では、次の処理が自動化されています。

- データベースのシャットダウン
- 既存のファイルシステムの読み取り専用状態への変換
- ソース・ファイル・システムからターゲット・ファイル・システムへのすべてのデータのコピー（すべてのファイル権限を保持）
- 古いファイルシステムと新しいファイルシステムのアンマウント
- 以前のファイルシステムと同じパスでの新しいファイルシステムの再マウント

## 手順

1. データベースをシャットダウンします。

```
[root@host1 current]# ./dbshut.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 15:58:48 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down
```

2. ファイルシステムを読み取り専用に変換します。スクリプトを使用すると、に示すように、この処理をより迅速に実行できます。 ["ファイルシステムを読み取り専用に変換"](#)。

```
[root@host1 current]# ./mk.fs.readonly.pl /oradata
/oradata unmounted
/oradata mounted read-only
[root@host1 current]# ./mk.fs.readonly.pl /logs
/logs unmounted
/logs mounted read-only
```

3. ファイルシステムが読み取り専用になったことを確認します。

```
ontap-nfs1:/host1_oradata on /oradata type nfs
(ro,bg,vers=3,rsz=65536,wsz=65536,addr=172.20.101.10)
ontap-nfs1:/host1_logs on /logs type nfs
(ro,bg,vers=3,rsz=65536,wsz=65536,addr=172.20.101.10)
```

4. ファイルシステムの内容を rsync コマンドを実行します

```
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /oradata/ /new/oradata/
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
```

```

10737426432 100% 153.50MB/s 0:01:06 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
22823573 100% 12.09MB/s 0:00:01 (xfer#2, to-check=9/13)
...
NTAP/undotbs02.dbf
1073750016 100% 131.60MB/s 0:00:07 (xfer#10, to-check=1/13)
NTAP/users01.dbf
5251072 100% 3.95MB/s 0:00:01 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes received 228 bytes 162204017.96 bytes/sec
total size is 18570092218 speedup is 1.00
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /logs/ /new/logs/
sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
45523968 100% 95.98MB/s 0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
40601088 100% 49.45MB/s 0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
52429312 100% 44.68MB/s 0:00:01 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
52429312 100% 68.03MB/s 0:00:00 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278

```

```
sent 527098156 bytes   received 278 bytes   95836078.91 bytes/sec
total size is 527032832   speedup is 1.00
```

5. 古いファイルシステムをアンマウントし、コピーしたデータを再配置します。スクリプトを使用すると、[示すように](#)、この処理をより迅速に実行できます。"[ファイルシステムの置き換え](#)"。

```
[root@host1 current]# ./swap.fs.pl /logs,/new/logs
/new/logs unmounted
/logs unmounted
Updated /logs mounted
[root@host1 current]# ./swap.fs.pl /oradata,/new/oradata
/new/oradata unmounted
/oradata unmounted
Updated /oradata mounted
```

6. 新しいファイルシステムが所定の位置にあることを確認します。

```
ontap-nfs1:/host1_logs_new on /logs type nfs
(rw,bg,vers=3,rsz=65536,wsz=65536,addr=172.20.101.10)
ontap-nfs1:/host1_oradata_new on /oradata type nfs
(rw,bg,vers=3,rsz=65536,wsz=65536,addr=172.20.101.10)
```

7. データベースを起動します。

```
[root@host1 current]# ./dbstart.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 16:10:07 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                 2929552 bytes
Variable Size              390073456 bytes
Database Buffers           406847488 bytes
Redo Buffers                5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
```

カットオーバーを完全に自動化

このサンプルスクリプトでは、データベースSIDの引数に続いて、共通区切りのファイルシステムペアを指定します。上記の例では、コマンドは次のように実行されます。

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs  
/oradata,/new/oradata
```

このサンプルスクリプトを実行すると、次のシーケンスが試行されます。いずれかの手順でエラーが発生すると終了します。

1. データベースをシャットダウンします。
2. 現在のファイルシステムを読み取り専用ステータスに変換します。
3. カンマで区切られた各ファイルシステム引数のペアを使用し、最初のファイルシステムを2番目のファイルシステムに同期します。
4. 以前のファイルシステムをディスマウントします。
5. を更新します /etc/fstab ファイルは次のとおりです。
  - a. バックアップの作成場所 /etc/fstab.bak。
  - b. 以前のファイルシステムと新しいファイルシステムの前のエントリをコメントアウトします。
  - c. 古いマウントポイントを使用する新しいファイルシステム用の新しいエントリを作成します。
6. ファイルシステムをマウントします。
7. データベースを起動します。

次のテキストは、このスクリプトの実行例を示しています。

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs  
/oradata,/new/oradata  
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin  
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:05:50 2015  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit  
Production  
With the Partitioning, OLAP, Advanced Analytics and Real Application  
Testing options  
SQL> Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release  
12.1.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real Application  
Testing options  
NTAP shut down
```

```

sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
    45523968 100%  185.40MB/s    0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
    40601088 100%   81.34MB/s    0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
    52429312 100%   70.42MB/s    0:00:00 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
    52429312 100%   47.08MB/s    0:00:01 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
sent 527098156 bytes  received 278 bytes  150599552.57 bytes/sec
total size is 527032832  speedup is 1.00
Succesfully replicated filesystem /logs to /new/logs
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
    10737426432 100%  176.55MB/s    0:00:58 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
    22823573 100%    9.48MB/s    0:00:02 (xfer#2, to-check=9/13)
... NTAP/undotbs01.dbf
    309338112 100%   70.76MB/s    0:00:04 (xfer#9, to-check=2/13)
NTAP/undotbs02.dbf
    1073750016 100%  187.65MB/s    0:00:05 (xfer#10, to-check=1/13)
NTAP/users01.dbf
    5251072 100%    5.09MB/s    0:00:00 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277

```



```

File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes received 228 bytes 177725933.55 bytes/sec
total size is 18570092218 speedup is 1.00
Succesfully replicated filesystem /oradata to /new/oradata
swap 0 /logs /new/logs
/new/logs unmounted
/logs unmounted
Mounted updated /logs
Swapped filesystem /logs for /new/logs
swap 1 /oradata /new/oradata
/new/oradata unmounted
/oradata unmounted
Mounted updated /oradata
Swapped filesystem /oradata for /new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:08:59 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area 805306368 bytes
Fixed Size 2929552 bytes
Variable Size 390073456 bytes
Database Buffers 406847488 bytes
Redo Buffers 5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
[root@host1 current]#

```

## Oracle ASM spfileとpasswdの移行

ASMを含む移行を完了する際の難しさの1つに、ASM固有のspfileとパスワードファイルがあります。デフォルトでは、これらの重要なメタデータファイルは、最初に定義されたASMディスクグループに作成されます。特定のASMディスクグループを退避して削除する必要がある場合は、そのASMインスタンスを制御するspfileファイルとパスワードファイルを再配置する必要があります。

これらのファイルの再配置が必要になる別のユースケースとして、SnapManager for OracleやSnapCenter Oracleプラグインなどのデータベース管理ソフトウェアを導入する場合があります。これらの製品の機能の1つは、データファイルをホストしているASM LUNの状態をリポートして、データベースを迅速にリストアすることです。そのためには、リストアを実行する前にASMディスクグループをオフラインにする必要があります。

ます。特定のデータベースのデータファイルが専用のASMディスクグループに分離されていれば、これは問題になりません。

そのディスクグループにASM spfile/passwdファイルも含まれている場合、ディスクグループをオフラインにするには、ASMインスタンス全体をシャットダウンするしかありません。これはシステムの停止を伴うプロセスであり、spfile/passwdファイルを再配置する必要があります。

#### 環境

1. データベースSID = トースト
2. 現在のデータファイル： +DATA
3. 現在のログファイルと制御ファイル +LOGS
4. シンシイASMディスククルウフノセツテイ +NEWDATA および +NEWLOGS

#### ASM spfile/passwdファイルの場所

これらのファイルは、システムを停止することなく再配置できます。ただし、安全のために、NetAppでは、ファイルが再配置され、構成が適切に更新されたことを確実に確認できるように、データベース環境をシャットダウンすることを推奨しています。サーバに複数のASMインスタンスが存在する場合は、この手順を繰り返す必要があります。

#### ASMインスタンスの識別

に記録されたデータに基づいてASMインスタンスを特定します。 oratab ファイル。ASMインスタンスは+記号で示されます。

```
-bash-4.1$ cat /etc/oratab | grep '^+'  
+ASM:/orabin/grid:N          # line added by Agent
```

このサーバには+asmというASMインスタンスが1つあります。

すべてのデータベースがシャットダウンされていることを確認する

表示されるSMONプロセスは、使用中のASMインスタンスのSMONだけです。別のSMONプロセスが存在する場合は、データベースが実行中であることを示します。

```
-bash-4.1$ ps -ef | grep smon  
oracle      857      1   0 18:26 ?          00:00:00 asm_smon_+ASM
```

SMONプロセスはASMインスタンス自体のみです。これは、他のデータベースが実行されていないことを意味し、データベースの処理を中断するリスクを伴わずに、安全に処理を続行できることを意味します。

#### ファイルの検索

次のコマンドを使用して、ASM spfileおよびパスワードファイルの現在の場所を特定します。 spget および pwget コマンド

```
bash-4.1$ asmcmd
ASMCMD> spget
+DATA/spfile.ora
```

```
ASMCMD> pwget --asm
+DATA/orapwasm
```

これらのファイルは両方とも、+DATA ディスクグループ：

ファイルのコピー

次のコマンドを使用して、ファイルを新しいASMディスクグループにコピーします。spcopy および pwcopy コマンド新しいディスクグループが最近作成され、現在空の場合は、最初にマウントする必要があります。

```
ASMCMD> mount NEWDATA
```

```
ASMCMD> spcopy +DATA/spfile.ora +NEWDATA/spfile.ora
copying +DATA/spfile.ora -> +NEWDATA/spfilea.ora
```

```
ASMCMD> pwcopy +DATA/orapwasm +NEWDATA/orapwasm
copying +DATA/orapwasm -> +NEWDATA/orapwasm
```

ファイルは次の場所からコピーされました：+DATA 終了：+NEWDATA。

## ASMインスタンスの更新

ASMインスタンスを更新して、場所の変更を反映する必要があります。。spset および pwset コマンドは、ASMディスクグループの起動に必要なASMメタデータを更新します。

```
ASMCMD> spset +NEWDATA/spfile.ora
ASMCMD> pwset --asm +NEWDATA/orapwasm
```

## 更新ファイルを使用したASMのアクティブ化

この時点で、ASMインスタンスは引き続きこれらのファイルの以前の場所を使用します。新しい場所からファイルを強制的に再読み込みし、以前のファイルのロックを解除するには、インスタンスを再起動する必要があります。

```
-bash-4.1$ sqlplus / as sysasm
SQL> shutdown immediate;
ASM diskgroups volume disabled
ASM diskgroups dismounted
ASM instance shutdown
```

```
SQL> startup
ASM instance started
Total System Global Area 1140850688 bytes
Fixed Size                2933400 bytes
Variable Size             1112751464 bytes
ASM Cache                 25165824 bytes
ORA-15032: not all alterations performed
ORA-15017: diskgroup "NEWDATA" cannot be mounted
ORA-15013: diskgroup "NEWDATA" is already mounted
```

### 古いspfileファイルとパスワードファイルを削除する

手順が正常に実行されると、以前のファイルはロックされなくなり、削除できるようになります。

```
-bash-4.1$ asmcmd
ASMCMD> rm +DATA/spfile.ora
ASMCMD> rm +DATA/orapwasm
```

### Oracle ASMからASMヘノコヒイ

Oracle ASMは、基本的に軽量なボリュームマネージャとファイルシステムを統合したものです。ファイルシステムはすぐには認識されないため、RMANを使用してコピー処理を実行する必要があります。コピーベースの移動プロセスは安全でシンプルですが、システム停止が発生することがあります。システム停止を最小限に抑えることはできますが、完全に排除することはできません。

ASMベースのデータベースを無停止で移行する場合は、ASMの機能を活用して、古いLUNを削除しながらASMエクステントを新しいLUNにリバランシングすることを推奨します。これは一般に安全でノンストップオペレーションですが、バックアウトパスは提供されません。機能またはパフォーマンスの問題が発生した場合、唯一の選択肢はデータをソースに戻すことです。

このリスクを回避するには、データを移動するのではなく、データベースを新しい場所にコピーして、元のデータに変更を加えないようにします。データベースは、稼働を開始する前に新しい場所で完全にテストすることができ、問題が見つかった場合は、元のデータベースをフォールバックオプションとして使用できます。

この手順は、RMANに関連する多数のオプションの1つです。最初のバックアップが作成され、ログ再生によって後で同期される2段階のプロセスが可能になります。このプロセスでは、最初のベースラインコピーの実行中もデータベースの運用を維持し、データを提供できるため、ダウンタイムを最小限に抑えることが推奨されます。

## データベースコピー

Oracle RMANは、ASMディスクグループに現在配置されているソースデータベースのレベル0（完全）コピーを作成します。+DATA 次の場所に移動します：+NEWDATA。

```
-bash-4.1$ rman target /
Recovery Manager: Release 12.1.0.2.0 - Production on Sun Dec 6 17:40:03
2015
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All rights
reserved.
connected to target database: TOAST (DBID=2084313411)
RMAN> backup as copy incremental level 0 database format '+NEWDATA' tag
'ONTAP_MIGRATION';
Starting backup at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=302 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
...
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
output file name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
tag=ONTAP_MIGRATION RECID=5 STAMP=897759622
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWDATA/TOAST/BACKUPSET/2015_12_06/nnsnn0_ontap_migration_0.262.89
7759623 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

## アーカイブログの強制切り替え

コピーの完全な整合性を確保するために必要なすべてのデータがアーカイブログに含まれていることを確認するには、アーカイブログを強制的に切り替えます。このコマンドを使用しないと、REDOログにキーデータが残っている可能性があります。

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

## ソースデータベースのシャットダウン

データベースがシャットダウンされ、アクセスが制限された読み取り専用モードになるため、システムが停止します。ソースデータベースをシャットダウンするには、次のコマンドを実行します。

```

RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area      805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                  390073456 bytes
Database Buffers               406847488 bytes
Redo Buffers                    5455872 bytes

```

## 制御ファイルのバックアップ

移行を中止して元のストレージの場所に戻す必要がある場合に備えて、制御ファイルをバックアップする必要があります。バックアップ制御ファイルのコピーは100%必要ではありませんが、データベースファイルの場所を元の場所にリセットする処理が簡単になります。

```

RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=358 device type=DISK
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151206T174753 RECID=6
STAMP=897760073
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15

```

## パラメータの更新

現在のspfileには、古いASMディスクグループ内の現在の場所にある制御ファイルへの参照が含まれています。編集する必要があります。これは、中間のpfileバージョンを編集することで簡単に実行できます。

```

RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed

```

## pfileの更新

古いASMディスクグループを参照しているすべてのパラメータを更新し、新しいASMディスクグループ名を反映させます。次に、更新されたpfileを保存します。次のことを確認します。 db\_create パラメータが存在します。

次の例では、 +DATA 変更されました +NEWDATA 黄色で強調表示されます。主なパラメータは次の2つです。 db\_create 正しい場所に新しいファイルを作成するパラメータ。

```
*.compatible='12.1.0.2.0'
*.control_files='+NEWLOGS/TOAST/CONTROLFILE/current.258.897683139'
*.db_block_size=8192
*. db_create_file_dest='+NEWDATA'
*. db_create_online_log_dest_1='+NEWLOGS'
*.db_domain=''
*.db_name='TOAST'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB) '
*.log_archive_dest_1='LOCATION='+NEWLOGS'
*.log_archive_format='%t_%s_%r.dbf'
```

## init.oraファイルの更新

ほとんどのASMベースのデータベースでは、 init.ora ファイルはにあります \$ORACLE\_HOME/dbs ディレクトリ。ASMディスクグループ上のspfileへのポイントです。このファイルは、新しいASMディスクグループの場所にリダイレクトする必要があります。

```
-bash-4.1$ cd $ORACLE_HOME/dbs
-bash-4.1$ cat initTOAST.ora
SPFILE='+DATA/TOAST/spfileTOAST.ora'
```

このファイルを次のように変更します。

```
SPFILE='+NEWLOGS/TOAST/spfileTOAST.ora'
```

## パラメータファイルの再作成

これで編集したpfileのデータをspfileに入力する準備が整いました

```
RMAN> create spfile from pfile='/tmp/pfile';
Statement processed
```

新しいspfileの使用を開始するには'データベースを起動します

データベースを起動して、新しく作成されたspfileが使用されていること、およびシステムパラメータに対するそれ以降の変更が正しく記録されていることを確認します。

```
RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area      805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                  373296240 bytes
Database Buffers               423624704 bytes
Redo Buffers                   5455872 bytes
```

制御ファイルのリストア

RMANによって作成されたバックアップ制御ファイルは、RMANによって、新しいspfileに指定された場所に直接リストアすることもできます。

```
RMAN> restore controlfile from
'+DATA/TOAST/CONTROLFILE/current.258.897683139';
Starting restore at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=417 device type=DISK
channel ORA_DISK_1: copied control file copy
output file name=+NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
Finished restore at 06-DEC-15
```

データベースをマウントし、新しい制御ファイルが使用されていることを確認します。

```
RMAN> alter database mount;
using target database control file instead of recovery catalog
Statement processed
```

```
SQL> show parameter control_files;
NAME                                TYPE                                VALUE
-----
control_files                       string
+NEWLOGS/TOAST/CONTROLFILE/cur
rent.273.897761061
```



## ログ再生

データベースは現在、古い場所にあるデータファイルを使用しています。コピーを使用する前に、コピーを同期する必要があります。最初のコピープロセスで時間が経過し、主にアーカイブログに変更が記録されました。これらの変更は次のように複製されます。

1. アーカイブ・ログを含むRMAN増分バックアップを実行します。

```
RMAN> backup incremental level 1 format '+NEWLOGS' for recover of copy
with tag 'ONTAP_MIGRATION' database;
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=62 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
input datafile file number=00002
name=+DATA/TOAST/DATAFILE/sysaux.260.897683143
input datafile file number=00003
name=+DATA/TOAST/DATAFILE/undotbs1.257.897683145
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/ncsnn1_ontap_migration_0.267.
897762697 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

2. ログを再生します。

```

RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 06-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001
name=+NEWDATA/TOAST/DATAFILE/system.259.897759609
recovering datafile copy file number=00002
name=+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
recovering datafile copy file number=00003
name=+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
recovering datafile copy file number=00004
name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
channel ORA_DISK_1: reading from backup piece
+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.8977626
93
channel ORA_DISK_1: piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15

```

## アクティブ化

リストアされた制御ファイルは元の場所にあるデータ・ファイルを参照しており、コピーされたデータ・ファイルのパス情報も含まれています。

1. アクティブなデータファイルを変更するには、switch database to copy コマンドを実行します

```

RMAN> switch database to copy;
datafile 1 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/system.259.897759609"
datafile 2 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615"
datafile 3 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619"
datafile 4 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/users.258.897759623"

```

アクティブなデータファイルがコピーされたデータファイルになりますが、最終的なREDOログに変更が含まれている可能性があります。

2. 残りのログをすべて再生するには、recover database コマンドを実行しますというメッセージが表示されます media recovery complete と表示され、プロセスは成功しました。

```

RMAN> recover database;
Starting recover at 06-DEC-15
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15

```

このプロセスで変更されるのは、通常のデータファイルの場所だけです。一時データファイルの名前は変更する必要がありますが、一時データファイルであるためコピーする必要はありません。データベースは現在ダウンしているため、一時データファイルにアクティブなデータはありません。

3. 一時データファイルを移動するには、まずその場所を特定します。

```

RMAN> select file#||' '||name from v$tempfile;
FILE#||' '||NAME
-----
1 +DATA/TOAST/TEMPFILE/temp.263.897683145

```

4. 各データファイルに新しい名前を設定するRMANコマンドを使用して、一時データファイルを移動します。Oracle Managed Files (OMF) では、完全な名前は必要ありません。ASMディスクグループで十分です。データベースが開くと、OMFはASMディスクグループ上の適切な場所にリンクします。ファイルを再配置するには、次のコマンドを実行します。

```

run {
set newname for tempfile 1 to '+NEWDATA';
switch tempfile all;
}

```

```

RMAN> run {
2> set newname for tempfile 1 to '+NEWDATA';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to +NEWDATA in control file

```

## Redoログの移行

移行プロセスはほぼ完了していますが、REDOログは元のASMディスクグループに残ります。REDOログは直接再配置できません。代わりに、新しいREDOログセットが作成されて設定に追加され、古いログがドロップされます。

1. REDOロググループの数とそれぞれのグループ番号を確認します。

```

RMAN> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 +DATA/TOAST/ONLINELOG/group_1.261.897683139
2 +DATA/TOAST/ONLINELOG/group_2.259.897683139
3 +DATA/TOAST/ONLINELOG/group_3.256.897683139

```

2. Redoログのサイズを入力します。

```

RMAN> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 52428800
2 52428800
3 52428800

```

3. Redoログごとに、設定が一致する新しいグループを作成します。OMFを使用しない場合は、フルパスを指定する必要があります。また、この例では、`db_create_online_log` パラメータ前述のように、このパラメータは+NEWLOGSに設定されています。この設定では、次のコマンドを使用して、ファイルの場所や特定のASMディスクグループを指定することなく、新しいオンラインログを作成できます。

```

RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed

```

4. データベースを開きます。

```

SQL> alter database open;
Database altered.

```

5. 古いログを削除します。

```

RMAN> alter database drop logfile group 1;
Statement processed

```

6. アクティブなログをドロップできないエラーが発生した場合は、次のログに切り替えてロックを解除し、

グローバルチェックポイントを強制的に実行します。以下に例を示します。古い場所にあるログファイルグループ3を削除しようとしたが、このログファイルにアクティブなデータが残っているため拒否されました。チェックポイントに続くログアーカイブでは、ログファイルを削除できます。

```
RMAN> alter database drop logfile group 3;
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 3 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 3 thread 1:
'+LOGS/TOAST/ONLINELOG/group_3.259.897563549'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 3;
Statement processed
```

7. 環境をレビューして、すべてのロケーションベースのパラメータが更新されていることを確認します。

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

8. 次のスクリプトは、このプロセスを簡素化する方法を示しています。

```
[root@host1 current]# ./checkdbdata.pl TOAST
TOAST datafiles:
+NEWDATA/TOAST/DATAFILE/system.259.897759609
+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
+NEWDATA/TOAST/DATAFILE/users.258.897759623
TOAST redo logs:
+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123
+NEWLOGS/TOAST/ONLINELOG/group_5.265.897763125
+NEWLOGS/TOAST/ONLINELOG/group_6.264.897763125
TOAST temp datafiles:
+NEWDATA/TOAST/TEMPFILE/temp.260.897763165
TOAST spfile
spfile                                string
+NEWDATA/spfiletoast.ora
TOAST key parameters
control_files +NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
log_archive_dest_1 LOCATION=+NEWLOGS
db_create_file_dest +NEWDATA
db_create_online_log_dest_1 +NEWLOGS
```

9. ASMディスクグループが完全に退避された場合は、次のコマンドを使用してアンマウントできます。  
asmcmd。ただし、多くの場合、他のデータベースまたはASM spfile/passwdファイルに属するファイルが存在する可能性があります。

```
-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMDS> umount DATA
ASMCMDS>
```

## Oracle ASMからファイルシステムへのコピー

Oracle ASMからファイルシステムへのコピー手順は、ASMからASMへのコピー手順と非常によく似ていますが、利点と制限は似ています。主な違いは、ASMディスクグループではなく可視ファイルシステムを使用する場合の、さまざまなコマンドや設定パラメータの構文です。

### データベースコピー

Oracle RMANを使用して、ASMディスクグループに現在配置されているソースデータベースのレベル0（完全）コピーを作成します。+DATA 次の場所に移動します： /oradata。

```

RMAN> backup as copy incremental level 0 database format
'/oradata/TOAST/%U' tag 'ONTAP_MIGRATION';
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=377 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-
1_01r5fhjg tag=ONTAP_MIGRATION RECID=1 STAMP=911722099
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-
2_02r5fhjo tag=ONTAP_MIGRATION RECID=2 STAMP=911722106
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt tag=ONTAP_MIGRATION RECID=3 STAMP=911722113
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/oradata/TOAST/cf_D-TOAST_id-2098173325_04r5fhk5
tag=ONTAP_MIGRATION RECID=4 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-
4_05r5fhk6 tag=ONTAP_MIGRATION RECID=5 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/oradata/TOAST/06r5fhk7_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16

```

#### アーカイブログの強制切り替え

コピーの完全な整合性を確保するために必要なすべてのデータがアーカイブログに含まれていることを確認するには、アーカイブログの切り替えを強制する必要があります。このコマンドを使用しないと、REDOログにキーデータが残っている可能性があります。アーカイブログを強制的に切り替えるには、次のコマンドを実行します。

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

#### ソースデータベースのシャットダウン

データベースがシャットダウンされ、アクセスが制限された読み取り専用モードになるため、システムが停止します。ソースデータベースをシャットダウンするには、次のコマンドを実行します。

```
RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area      805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                  331353200 bytes
Database Buffers               465567744 bytes
Redo Buffers                    5455872 bytes
```

#### 制御ファイルのバックアップ

移行を中止して元のストレージの場所に戻す必要がある場合に備えて、制御ファイルをバックアップします。バックアップ制御ファイルのコピーは100%必要ではありませんが、データベースファイルの場所を元の場所にリセットする処理が簡単になります。

```
RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 08-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151208T194540 RECID=30
STAMP=897939940
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 08-DEC-15
```

#### パラメータの更新

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```



## pfileの更新

古いASMディスクグループを参照するすべてのパラメータは、関連性がなくなったときに更新し、場合によっては削除する必要があります。新しいファイルシステムパスを反映するように更新し、更新されたpfileを保存します。完全なターゲットパスが表示されていることを確認します。これらのパラメータを更新するには、次のコマンドを実行します。

```
*.audit_file_dest='/orabin/admin/TOAST/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/logs/TOAST/arch/control01.ctl','/logs/TOAST/redo/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='TOAST'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB) '
*.log_archive_dest_1='LOCATION=/logs/TOAST/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

## 元のinit.oraファイルを無効にする

このファイルは、\$ORACLE\_HOME/dbs ディレクトリとは、通常、ASMディスクグループ上のspfileへのポインタとして機能するpfile内にあります。元のspfileが使用されていないことを確認するには、名前を変更します。ただし、このファイルは移行を中止する必要がある場合に必要になるため、削除しないでください。

```
[oracle@jfscl ~]$ cd $ORACLE_HOME/dbs
[oracle@jfscl dbs]$ cat initTOAST.ora
SPFILE='+ASM0/TOAST/spfileTOAST.ora'
[oracle@jfscl dbs]$ mv initTOAST.ora initTOAST.ora.prev
[oracle@jfscl dbs]$
```

## パラメータファイルの再作成

これは'spfile再配置の最後の手順です元のspfileは使用されなくなり'中間ファイルを使用してデータベースが現在起動されています（マウントされていません）このファイルの内容は'次のようにして新しいspfileの場所に書き出すことができます

```
RMAN> create spfile from pfile='/tmp/pfile';  
Statement processed
```

新しい**spfile**の使用を開始するには'データベースを起動します

中間ファイルのロックを解除するには、データベースを起動し、新しいspfileファイルのみを使用してデータベースを起動する必要があります。データベースを起動すると、新しいspfileの場所が正しいことと、そのデータが有効であることも証明されます。

```
RMAN> shutdown immediate;  
Oracle instance shut down  
RMAN> startup nomount;  
connected to target database (not started)  
Oracle instance started  
Total System Global Area      805306368 bytes  
Fixed Size                     2929552 bytes  
Variable Size                  331353200 bytes  
Database Buffers               465567744 bytes  
Redo Buffers                    5455872 bytes
```

制御ファイルのリストア

バックアップ制御ファイルがパスに作成されました /tmp/TOAST.ctrl 手順の初期段階。新しいspfileでは、制御ファイルの場所を次のように定義します。 /logfs/TOAST/ctrl/ctrlfile1.ctrl および /logfs/TOAST/redo/ctrlfile2.ctrl。ただし、これらのファイルはまだ存在しません。

1. このコマンドは、spfileに定義されているパスに制御ファイルのデータをリストアします。

```
RMAN> restore controlfile from '/tmp/TOAST.ctrl';  
Starting restore at 13-MAY-16  
using channel ORA_DISK_1  
channel ORA_DISK_1: copied control file copy  
output file name=/logs/TOAST/arch/control01.ctrl  
output file name=/logs/TOAST/redo/control02.ctrl  
Finished restore at 13-MAY-16
```

2. mountコマンドを問題して、制御ファイルが正しく検出され、有効なデータが含まれていることを確認します。

```
RMAN> alter database mount;  
Statement processed  
released channel: ORA_DISK_1
```

を検証するには control\_files パラメータを指定して、次のコマンドを実行します。

```
SQL> show parameter control_files;
NAME                                TYPE                                VALUE
-----                                -
control_files                       string
/logs/TOAST/arch/control01.ctl
                                     '
/logs/TOAST/redo/control02.c
                                     tl
```

## ログ再生

データベースは現在、古い場所にあるデータファイルを使用しています。コピーを使用する前に、データファイルを同期する必要があります。最初のコピープロセスで時間が経過し、変更は主にアーカイブログに記録されました。これらの変更は、次の2つのステップで複製されます。

1. アーカイブ・ログを含むRMAN増分バックアップを実行します。

```
RMAN> backup incremental level 1 format '/logs/TOAST/arch/%U' for
recover of copy with tag 'ONTAP_MIGRATION' database;
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=124 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/logs/TOAST/arch/09r5fj8i_1_1 tag=ONTAP_MIGRATION
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped
```

2. ログを再生します。

```

RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 13-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
recovering datafile copy file number=00002 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
recovering datafile copy file number=00003 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
recovering datafile copy file number=00004 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
channel ORA_DISK_1: reading from backup piece
/logs/TOAST/arch/09r5fj8i_1_1
channel ORA_DISK_1: piece handle=/logs/TOAST/arch/09r5fj8i_1_1
tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped

```

## アクティブ化

リストアされた制御ファイルは元の場所にあるデータ・ファイルを参照しており、コピーされたデータ・ファイルのパス情報も含まれています。

1. アクティブなデータファイルを変更するには、switch database to copy コマンドを実行します

```

RMAN> switch database to copy;
datafile 1 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSTEM_FNO-1_01r5fhjg"
datafile 2 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSAUX_FNO-2_02r5fhjo"
datafile 3 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt"
datafile 4 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-USERS_FNO-4_05r5fhk6"

```

2. データファイルの整合性は完全である必要がありますが、オンラインREDOログに記録された残りの変更を再生するには、最後に1つの手順を実行する必要があります。を使用します recover database これらの変更を再生し、コピーを元のコピーと100%同一にするコマンド。ただし、コピーはまだ開いていません。

```

RMAN> recover database;
Starting recover at 13-MAY-16
using channel ORA_DISK_1
starting media recovery
archived log for thread 1 with sequence 28 is already on disk as file
+ASM0/TOAST/redo01.log
archived log file name=+ASM0/TOAST/redo01.log thread=1 sequence=28
media recovery complete, elapsed time: 00:00:00
Finished recover at 13-MAY-16

```

## 一時データファイルの再配置

1. 元のディスクグループでまだ使用されている一時データファイルの場所を特定します。

```

RMAN> select file#||' '||name from v$tempfile;
FILE#||' '||NAME
-----
1 +ASM0/TOAST/temp01.dbf

```

2. データファイルを移動するには、次のコマンドを実行します。一時ファイルが多数ある場合は、テキスト・エディタを使用してRMANコマンドを作成し、それをカットアンドペーストします。

```

RMAN> run {
2> set newname for tempfile 1 to '/oradata/TOAST/temp01.dbf';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/TOAST/temp01.dbf in control file

```

## Redoログの移行

移行プロセスはほぼ完了していますが、REDOログは元のASMディスクグループに残ります。REDOログは直接再配置できません。代わりに、新しいREDOログセットが作成され、古いログがドロップされて設定に追加されます。

1. REDOロググループの数とそれぞれのグループ番号を確認します。

```

RMAN> select group#||' '||member from v$logfile;
GROUP#||' '||MEMBER
-----
-----
1 +ASM0/TOAST/redo01.log
2 +ASM0/TOAST/redo02.log
3 +ASM0/TOAST/redo03.log

```

2. Redoログのサイズを入力します。

```

RMAN> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 52428800
2 52428800
3 52428800

```

3. Redoログごとに、新しいファイルシステムの場所を使用して、現在のRedoロググループと同じサイズを使用して新しいグループを作成します。

```

RMAN> alter database add logfile '/logs/TOAST/redo/log00.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log01.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log02.rdo' size
52428800;
Statement processed

```

4. 以前のストレージにまだ配置されている古いログファイルグループを削除します。

```

RMAN> alter database drop logfile group 4;
Statement processed
RMAN> alter database drop logfile group 5;
Statement processed
RMAN> alter database drop logfile group 6;
Statement processed

```

5. アクティブログのドロップをブロックするエラーが発生した場合は、次のログに強制的に切り替えてロックを解除し、グローバルチェックポイントを強制的に実行します。以下に例を示します。古い場所にある

ログファイルグループ3を削除しようとしたが、このログファイルにアクティブなデータが残っているため拒否されました。ログをアーカイブしたあとにチェックポイントを追加すると、ログファイルの削除が可能になります。

```
RMAN> alter database drop logfile group 4;
RMAN-00571: =====
RMAN-00569: ===== ERROR MESSAGE STACK FOLLOWS =====
RMAN-00571: =====
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 4 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 4 thread 1:
'+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 4;
Statement processed
```

6. 環境をレビューして、すべてのロケーションベースのパラメータが更新されていることを確認します。

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

7. 次のスクリプトは、このプロセスを簡単にする方法を示しています。

```
[root@jfscl current]# ./checkdbdata.pl TOAST
TOAST datafiles:
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
TOAST redo logs:
/logs/TOAST/redo/log00.rdo
/logs/TOAST/redo/log01.rdo
/logs/TOAST/redo/log02.rdo
TOAST temp datafiles:
/oradata/TOAST/temp01.dbf
TOAST spfile
spfile                                string
/orabin/product/12.1.0/dbhome_
                                         1/dbs/spfileTOAST.ora

TOAST key parameters
control_files /logs/TOAST/arch/control01.ctl,
/logs/TOAST/redo/control02.ctl
log_archive_dest_1 LOCATION=/logs/TOAST/arch
```

8. ASMディスクグループが完全に退避された場合は、次のコマンドを使用してアンマウントできます。  
asmcmd。多くの場合、他のデータベースまたはASM spfile/passwdファイルに属するファイルは引き続き存在する可能性があります。

```
-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMDB> umount DATA
ASMCMDB>
```

#### データファイルのクリーンアップ手順

Oracle RMANの使用方法によっては、移行プロセスの結果、構文が長いデータファイルや暗号化されたデータファイルが生成されることがあります。この例では、次のファイル形式でバックアップが実行されています：  
/oradata/TOAST/%U。%U RMANが各データ・ファイルにデフォルトの一意の名前を作成する必要がありますことを示します。結果は次のテキストに示されているものと似ています。データファイルの従来の名前は、名前の中に埋め込まれています。これは、に示すスクリプト化されたアプローチを使用してクリーンアップできます。"[ASM移行クリーンアップ](#)"。



```
[root@jfscl current]# ./fixuniquenames.pl TOAST
#sqlplus Commands
shutdown immediate;
startup mount;
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/system.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/sysaux.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt /oradata/TOAST/undotbs1.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
/oradata/TOAST/users.dbf
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSTEM_FNO-1_01r5fhjg' to '/oradata/TOAST/system.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSAUX_FNO-2_02r5fhjo' to '/oradata/TOAST/sysaux.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
UNDOTBS1_FNO-3_03r5fhjt' to '/oradata/TOAST/undotbs1.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
USERS_FNO-4_05r5fhk6' to '/oradata/TOAST/users.dbf';
alter database open;
```

## Oracle ASMのリバランシング

前述したように、Oracle ASMディスクグループは、リバランシングプロセスを使用して新しいストレージシステムに透過的に移行できます。つまり、リバランシングプロセスでは、既存のLUNグループに同じサイズのLUNを追加してから、前のLUNを破棄する必要があります。Oracle ASMは、基盤となるデータを最適なレイアウトで新しいストレージに自動的に再配置し、完了すると古いLUNを解放します。

マイグレーションプロセスでは効率的なシーケンシャルI/Oを使用し、通常は原因パフォーマンスの中断は発生しませんが、必要に応じてマイグレーション速度を調整できます。

移行するデータを特定

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
SQL> select group_number||' '||name from v$asm_diskgroup;
1 NEWDATA
```

## 新しいLUNを作成する

同じサイズの新しいLUNを作成し、必要に応じてユーザとグループのメンバーシップを設定します。LUNはと表示されます。 CANDIDATE ディスク：

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||' '||header_status from v$asm_disk;
0 0 /dev/mapper/3600a098038303537762b47594c31586b CANDIDATE
0 0 /dev/mapper/3600a098038303537762b47594c315869 CANDIDATE
0 0 /dev/mapper/3600a098038303537762b47594c315858 CANDIDATE
0 0 /dev/mapper/3600a098038303537762b47594c31586a CANDIDATE
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
```

## 新しいLUNの追加

追加処理と削除処理は同時に実行できますが、新しいLUNを追加する方が2つの手順で簡単に実行できます。まず、新しいLUNをディスクグループに追加します。この手順により、エクステントの半分が現在のASM LUNから新しいLUNに移行されます。

リバランシング電力は、データが転送される速度を示します。数値が大きいほど、データ転送の並列性が高くなります。移行は、効率的なシーケンシャルI/O処理を使用して実行されますが、原因のパフォーマンスに問題が生じることはほとんどありません。ただし、必要に応じて、進行中の移行のリバランシング機能を `alter diskgroup [name] rebalance power [level]` コマンドを実行します。一般的な移行では、値5が使用されます。

```
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586b' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315869' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315858' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586a' rebalance power 5;
Diskgroup altered.
```

## 動作の監視

リバランシング処理は、さまざまな方法で監視および管理できます。この例では、次のコマンドを使用しました。

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
-----
1 REBAL RUN
1 REBAL WAIT
```

移行が完了しても、リバランシング処理は報告されません。

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

古いLUNを削除する

移行は途中で完了しました。環境が健全であることを確認するために、いくつかの基本的なパフォーマンステストを実行することを推奨します。確認後、古いLUNを削除して残りのデータを再配置できます。これによってLUNがすぐに解放されるわけではないことに注意してください。drop処理は、最初にエクステントを再配置してからLUNを解放するようOracle ASMに通知します。

```
sqlplus / as sysasm
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0000 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0001 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0002 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0003 rebalance power 5;
Diskgroup altered.
```

動作の監視

リバランシング処理は、さまざまな方法で監視および管理できます。この例では、次のコマンドを使用しました。

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
-----
1 REBAL RUN
1 REBAL WAIT
```

移行が完了しても、リバランシング処理は報告されません。

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

## 古いLUNを削除する

ディスクグループから古いLUNを削除する前に、ヘッダーのステータスを最後に確認する必要があります。ASMからLUNを解放すると、LUNの名前は表示されなくなり、ヘッダーステータスが FORMER。これは、これらのLUNをシステムから安全に削除できることを示します。

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NAME||' '||GROUP_NUMBER||' '||TOTAL_MB||' '||PATH||' '||HEADER_STATUS
-----
-----
0 0 /dev/mapper/3600a098038303537762b47594c315863 FORMER
0 0 /dev/mapper/3600a098038303537762b47594c315864 FORMER
0 0 /dev/mapper/3600a098038303537762b47594c315861 FORMER
0 0 /dev/mapper/3600a098038303537762b47594c315862 FORMER
NEWDATA_0005 1 10240 /dev/mapper/3600a098038303537762b47594c315869 MEMBER
NEWDATA_0007 1 10240 /dev/mapper/3600a098038303537762b47594c31586a MEMBER
NEWDATA_0004 1 10240 /dev/mapper/3600a098038303537762b47594c31586b MEMBER
NEWDATA_0006 1 10240 /dev/mapper/3600a098038303537762b47594c315858 MEMBER
8 rows selected.
```

## LVMの移行

ここに示す手順は、LVMベースのボリュームグループ移動の原則を示しています。datavg。これらの例はLinux LVMを参考にしていますが、原則はAIX、HP-UX、VxVMにも当てはまります。正確なコマンドは異なる場合があります。

1. 現在に含まれているLUNを特定します。 datavg ボリュームグループ：

```
[root@host1 ~]# pvdisplay -C | grep datavg
/dev/mapper/3600a098038303537762b47594c31582f datavg lvm2 a-- 10.00g
10.00g
/dev/mapper/3600a098038303537762b47594c31585a datavg lvm2 a-- 10.00g
10.00g
/dev/mapper/3600a098038303537762b47594c315859 datavg lvm2 a-- 10.00g
10.00g
/dev/mapper/3600a098038303537762b47594c31586c datavg lvm2 a-- 10.00g
10.00g
```

2. 物理サイズが同じか少し大きい新しいLUNを作成し、物理ボリュームとして定義します。

```
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315864
Physical volume "/dev/mapper/3600a098038303537762b47594c315864"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315863
Physical volume "/dev/mapper/3600a098038303537762b47594c315863"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315862
Physical volume "/dev/mapper/3600a098038303537762b47594c315862"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315861
Physical volume "/dev/mapper/3600a098038303537762b47594c315861"
successfully created
```

3. 新しいボリュームをボリュームグループに追加します。

```
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315864
Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315863
Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315862
Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315861
Volume group "datavg" successfully extended
```

4. 問題 pvmove コマンドを使用して、現在の各LUNのエクステンツを新しいLUNに再配置します。。 - i  
[seconds] 引数は、操作の進行状況を監視します。

```

[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31582f
/dev/mapper/3600a098038303537762b47594c315864
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 14.2%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 28.4%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 42.5%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 57.1%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 72.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 87.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31585a
/dev/mapper/3600a098038303537762b47594c315863
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 14.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 29.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 44.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 60.1%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 75.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 90.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c315859
/dev/mapper/3600a098038303537762b47594c315862
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 14.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 29.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 45.5%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 61.1%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 76.6%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 91.7%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31586c
/dev/mapper/3600a098038303537762b47594c315861
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 15.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 30.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 46.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 61.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 77.2%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 92.3%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 100.0%

```

5. このプロセスが完了したら、`vgreduce` コマンドを実行します成功すると、LUNをシステムから安全に削除できるようになります。

```
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31582f
Removed "/dev/mapper/3600a098038303537762b47594c31582f" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31585a
Removed "/dev/mapper/3600a098038303537762b47594c31585a" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c315859
Removed "/dev/mapper/3600a098038303537762b47594c315859" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31586c
Removed "/dev/mapper/3600a098038303537762b47594c31586c" from volume
group "datavg"
```

## ForeignLUNImport

### 計画

FLIを使用してSANリソースを移行する手順については、NetAppを参照して ["ONTAP Foreign LUN Import に関するドキュメント"](#) ください。

データベースとホストの観点からは、特別な手順は必要ありません。FCゾーンが更新されてLUNがONTAPで使用可能になると、LVMはLUNからLVMメタデータを読み取れるようになります。また、ボリュームグループを使用するための準備が整い、それ以上の設定手順は必要ありません。まれに、以前のストレージレイへの参照がハードコーディングされた構成ファイルが環境に含まれることがあります。例えばLinuxシステムには `/etc/multipath.conf` 特定のデバイスのWWNを参照するルールは、FLIで導入された変更を反映するように更新する必要があります。



サポートされている構成については、NetApp互換性マトリックスを参照してください。お使いの環境が含まれていない場合は、NetAppの担当者にお問い合わせください。

この例は、LinuxサーバでホストされているASM LUNとLVM LUNの両方の移行を示しています。FLIは他のオペレーティングシステムでもサポートされており、ホスト側のコマンドは異なる場合がありますが、原則は同じで、ONTAPの手順も同じです。

### LVM LUNの特定

準備の最初の手順は、移行するLUNを特定することです。この例では、2つのSANベースのファイルシステムが `/orabin` および `/backups`。

```
[root@host1 ~]# df -k
```

Filesystem	1K-blocks	Used	Available	Use%	
Mounted on					
/dev/mapper/rhel-root	52403200	8811464	43591736	17%	/
devtmpfs	65882776	0	65882776	0%	/dev
...					
fas8060-nfs-public:/install	199229440	119368128	79861312	60%	
/install					
/dev/mapper/sanvg-lvorabin	20961280	12348476	8612804	59%	
/orabin					
/dev/mapper/sanvg-lvbackups	73364480	62947536	10416944	86%	
/backups					

ボリューム・グループの名前は'（ボリューム・グループ名） - （論理ボリューム名）という形式のデバイス名から抽出できますこの場合、ボリュームグループの名前は sanvg。

。 pvdisplay このボリュームグループをサポートするLUNを特定するには、コマンドを次のように使用します。この例では、 sanvg ボリュームグループ：

```
[root@host1 ~]# pvdisplay -C -o pv_name,pv_size,pv_fmt,vg_name
```

PV	PSize	VG
/dev/mapper/3600a0980383030445424487556574266	10.00g	sanvg
/dev/mapper/3600a0980383030445424487556574267	10.00g	sanvg
/dev/mapper/3600a0980383030445424487556574268	10.00g	sanvg
/dev/mapper/3600a0980383030445424487556574269	10.00g	sanvg
/dev/mapper/3600a098038303044542448755657426a	10.00g	sanvg
/dev/mapper/3600a098038303044542448755657426b	10.00g	sanvg
/dev/mapper/3600a098038303044542448755657426c	10.00g	sanvg
/dev/mapper/3600a098038303044542448755657426d	10.00g	sanvg
/dev/mapper/3600a098038303044542448755657426e	10.00g	sanvg
/dev/mapper/3600a098038303044542448755657426f	10.00g	sanvg
/dev/sda2	278.38g	rhel

## ASM LUNの識別

ASM LUNも移行する必要があります。LUNとLUNパスの数をsqlplusからSYSASMユーザとして取得するには、次のコマンドを実行します。



```
SQL> select path||' '||os_mb from v$asm_disk;
PATH||' '||OS_MB
-----
-----
/dev/oracleasm/disks/ASM0 10240
/dev/oracleasm/disks/ASM9 10240
/dev/oracleasm/disks/ASM8 10240
/dev/oracleasm/disks/ASM7 10240
/dev/oracleasm/disks/ASM6 10240
/dev/oracleasm/disks/ASM5 10240
/dev/oracleasm/disks/ASM4 10240
/dev/oracleasm/disks/ASM1 10240
/dev/oracleasm/disks/ASM3 10240
/dev/oracleasm/disks/ASM2 10240
10 rows selected.
SQL>
```

## FCネットワークの変更

現在の環境には、移行するLUNが20個含まれています。現在のSANを更新して、ONTAPが現在のLUNにアクセスできるようにします。データはまだ移行されていませんが、ONTAPは現在のLUNから構成情報を読み取って、そのデータの新しいホームを作成する必要があります。

AFF / FASシステムの少なくとも1つのHBAポートをイニシエータポートとして設定する必要があります。また、ONTAPが外部ストレージアレイ上のLUNにアクセスできるように、FCゾーンを更新する必要があります。一部のストレージアレイでは、特定のLUNにアクセスできるWWNを制限するLUNマスキングが設定されています。その場合は、LUNマスキングも更新して、ONTAP WWNへのアクセスを許可する必要があります。

この手順が完了すると、ONTAPは外部ストレージアレイを `storage array show` コマンドを実行します返されるキーフィールドは、システム上の外部LUNの識別に使用されるプレフィックスです。次の例では、外部アレイ上のLUN `FOREIGN_1` プレフィックスを使用してONTAP内に表示されます。FOR-1。

## 外部アレイの識別

```
Cluster01::> storage array show -fields name,prefix
name          prefix
-----
FOREIGN_1     FOR-1
Cluster01::>
```

## 外部LUNの識別

LUNを表示するには、`array-name` に移動します `storage disk show` コマンドを実行します返されるデータは、移行手順中に複数回参照されます。

```
Cluster01::> storage disk show -array-name FOREIGN_1 -fields disk,serial
disk      serial-number
-----
FOR-1.1   800DT$HuVWBX
FOR-1.2   800DT$HuVWBZ
FOR-1.3   800DT$HuVWBW
FOR-1.4   800DT$HuVWBY
FOR-1.5   800DT$HuVWB/
FOR-1.6   800DT$HuVWBa
FOR-1.7   800DT$HuVWBd
FOR-1.8   800DT$HuVWBb
FOR-1.9   800DT$HuVWBc
FOR-1.10  800DT$HuVWBe
FOR-1.11  800DT$HuVWBf
FOR-1.12  800DT$HuVWBg
FOR-1.13  800DT$HuVWBh
FOR-1.14  800DT$HuVWBh
FOR-1.15  800DT$HuVWBj
FOR-1.16  800DT$HuVWBk
FOR-1.17  800DT$HuVWBm
FOR-1.18  800DT$HuVWBn
FOR-1.19  800DT$HuVWBp
FOR-1.20  800DT$HuVWBq
20 entries were displayed.
Cluster01::>
```

外部アレイ**LUN**をインポート候補として登録

外部LUNは、最初は特定のLUNタイプとして分類されます。データをインポートする前に、LUNを外部としてタグ付けする必要があるため、インポートプロセスの候補になる必要があります。この手順は、シリアル番号を `storage disk modify` 次の例に示すように、コマンドを実行します。このプロセスでは、ONTAP内でLUNのみが外部としてタグ付けされることに注意してください。外部LUN自体にはデータは書き込まれません。

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign true
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBp} -is
-foreign true
Cluster01::*>
```

## 移行したLUNをホストするボリュームの作成

移行したLUNをホストするにはボリュームが必要です。正確なボリューム構成は、ONTAPの機能を活用する全体的な計画によって異なります。この例では、ASM LUNが1つのボリュームに配置され、LVM LUNが2つ目のボリュームに配置されています。これにより、階層化、Snapshotの作成、QoS制御の設定などの目的で、LUNを独立したグループとして管理できます。

を設定します snapshot-policy `to` none。移行プロセスには、大量のデータの入れ替えが含まれる場合があります。そのため、Snapshotに不要なデータがキャプチャされるために誤ってSnapshotを作成すると、スペース消費が大幅に増加する可能性があります。

```
Cluster01::> volume create -volume new_asm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1152] Job succeeded: Successful
Cluster01::> volume create -volume new_lvm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1153] Job succeeded: Successful
Cluster01::>
```

## ONTAP LUNの作成

ボリュームを作成したら、新しいLUNを作成する必要があります。通常、LUNを作成する際にはLUNサイズなどの情報を指定する必要がありますが、この場合はforeign-disk引数がコマンドに渡されます。その結果、ONTAPは指定されたシリアル番号から現在のLUN設定データを複製します。また、LUNジオメトリとパーティションテーブルのデータを使用してLUNのアライメントを調整し、最適なパフォーマンスを確立します。

この手順では、外部アレイに対してシリアル番号を相互参照して、正しい外部LUNが正しい新しいLUNに照合されるようにする必要があります。

```
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_asm/LUN0 -ostype
linux -foreign-disk 800DT$HuVWBW
Created a LUN of size 10g (10737418240)
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_asm/LUN1 -ostype
linux -foreign-disk 800DT$HuVWBX
Created a LUN of size 10g (10737418240)
...
Created a LUN of size 10g (10737418240)
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_lvm/LUN8 -ostype
linux -foreign-disk 800DT$HuVWBn
Created a LUN of size 10g (10737418240)
Cluster01::~*> lun create -vserver vserver1 -path /vol/new_lvm/LUN9 -ostype
linux -foreign-disk 800DT$HuVWBo
Created a LUN of size 10g (10737418240)
```

## インポート関係を作成する

LUNは作成されましたが、レプリケーション先としては設定されていません。この手順を実行する前に、LUNをオフラインにする必要があります。この追加手順は、ユーザエラーからデータを保護するように設計されています。ONTAPでオンラインのLUNで移行を実行できると、入力ミスが原因でアクティブなデータが上書きされるリスクがあります。ユーザに最初にLUNをオフラインにするよう強制する追加手順は、正しいターゲットLUNが移行先として使用されていることを確認するのに役立ちます。

```
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN0
Warning: This command will take LUN "/vol/new_asm/LUN0" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN1
Warning: This command will take LUN "/vol/new_asm/LUN1" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
...
Warning: This command will take LUN "/vol/new_lvm/LUN8" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_lvm/LUN9
Warning: This command will take LUN "/vol/new_lvm/LUN9" in Vserver
        "vserver1" offline.
Do you want to continue? {y|n}: y
```

LUNがオフラインになったら、外部LUNのシリアル番号を `lun import create` コマンドを実行します

```
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN0
-foreign-disk 800DT$HuVWBW
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN1
-foreign-disk 800DT$HuVWBX
...
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN8
-foreign-disk 800DT$HuVWBn
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN9
-foreign-disk 800DT$HuVWBo
Cluster01::*>
```

すべてのインポート関係が確立されたら、LUNをオンラインに戻すことができます。

```
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN9
Cluster01::*>
```

## イニシエータグループの作成

イニシエータグループ (igroup) は、ONTAP LUNマスキングアーキテクチャの一部です。新しく作成したLUNには、ホストに最初にアクセスを許可しないかぎりアクセスできません。そのためには、アクセスを許可するFC WWNまたはiSCSIイニシエータ名をリストするigroupを作成します。このレポートの作成時点では、FLIはFC LUNでのみサポートされていました。ただし、移行後のiSCSIへの変換は簡単です（を参照）。["プロトコル変換"](#)。

この例では、ホストのHBAで使用可能な2つのポートに対応する2つのWWNを含むigroupが作成されます。

```
Cluster01::*> igroup create linuxhost -protocol fcp -ostype linux
-initiator 21:00:00:0e:1e:16:63:50 21:00:00:0e:1e:16:63:51
```

## 新しいLUNをホストにマッピング

igroupの作成後、LUNは定義したigroupにマッピングされます。これらのLUNは、このigroupに含まれるWWNでのみ使用できます。NetAppでは、移動プロセスのこの段階で、ホストがONTAPにゾーニングされていないことを前提としています。これは重要なことです。ホストが外部アレイと新しいONTAPシステムに同時にゾーニングされていると、各アレイで同じシリアル番号のLUNが検出されるリスクがあるためです。マルチパスの誤動作やデータの破損が発生する可能性があります。

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
Cluster01::*>
```

## カットオーバー

FCネットワーク設定を変更する必要があるため、Foreign LUN Importの実行中にシステムが一部停止することは避けられません。ただし、システム停止は、データベース環境を再起動してFCゾーニングを更新し、ホストのFC接続を外部LUNからONTAPに切り替えるために必要な時間よりもはるかに長く続く必要はありません。

このプロセスは次のように要約できます。

1. 外部LUN上のすべてのLUNアクティビティを休止します。
2. ホストのFC接続を新しいONTAPシステムにリダイレクトします。
3. インポートプロセスをトリガーします。
4. LUNを再検出します。
5. データベースを再起動します。

移行プロセスが完了するまで待つ必要はありません。特定のLUNの移行を開始すると、そのLUNをONTAPで使えるようになり、データコピープロセスを続行しながらデータを提供できます。すべての読み取りが外部LUNに渡され、すべての書き込みが両方のアレイに同期的に書き込まれます。コピー処理は非常に高速で、FCトラフィックのリダイレクトによるオーバーヘッドも最小限であるため、パフォーマンスへの影響は一時的で最小限に抑えてください。懸念事項がある場合は、移行プロセスが完了してインポート関係が削除されるまで、環境の再起動を遅らせることができます。

#### データベースをシャットダウン

この例の環境を休止する最初の手順は、データベースをシャットダウンすることです。

```
[oracle@host1 bin]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base remains unchanged with value /orabin
[oracle@host1 bin]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics
and Real Application Testing options
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

#### グリッドサービスをシャットダウン

移行するSANベースのファイルシステムの1つには、Oracle ASMサービスも含まれています。基盤となるLUNを休止するには、ファイルシステムをディスマウントする必要があります。つまり、このファイルシステム上で開いているファイルを含むプロセスをすべて停止する必要があります。

```
[oracle@host1 bin]$ ./crsctl stop has -f
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'host1'
CRS-2673: Attempting to stop 'ora.evmd' on 'host1'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host1'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host1'
CRS-2677: Stop of 'ora.DATA.dg' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host1'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host1' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host1' succeeded
CRS-2677: Stop of 'ora.asm' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host1'
CRS-2677: Stop of 'ora.cssd' on 'host1' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'host1' has completed
CRS-4133: Oracle High Availability Services has been stopped.
[oracle@host1 bin]$
```

#### ファイルシステムのディスマウント

すべてのプロセスがシャットダウンされると、アンマウント処理は成功します。権限が拒否された場合は、ファイルシステムがロックされているプロセスが存在する必要があります。。 fuser コマンドは、これらのプロセスを識別するのに役立ちます。

```
[root@host1 ~]# umount /orabin
[root@host1 ~]# umount /backups
```

#### ボリュームグループの非アクティブ化

特定のボリュームグループ内のすべてのファイルシステムがディスマウントされたら、そのボリュームグループを非アクティブ化できます。

```
[root@host1 ~]# vgchange --activate n sanvg
  0 logical volume(s) in volume group "sanvg" now active
[root@host1 ~]#
```

#### FCネットワークの変更

FCゾーンを更新して、ホストから外部アレイへのすべてのアクセスを削除し、ONTAPへのアクセスを確立できるようにしました。

#### インポートプロセスの開始

LUNインポートプロセスを開始するには、lun import start コマンドを実行します

```
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN0
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN1
...
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN8
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN9
Cluster01::lun import*>
```

#### インポートの進捗状況の監視

インポート操作を監視するには、`lun import show` コマンドを実行します次の図に示すように、20個すべてのLUNのインポートを実行中です。つまり、データコピー処理がまだ進行中であっても、ONTAPからデータにアクセスできるようになります。

```
Cluster01::lun import*> lun import show -fields path,percent-complete
vserver    foreign-disk path                                percent-complete
-----
vserver1   800DT$HuVWB/ /vol/new_asm/LUN4 5
vserver1   800DT$HuVWBW /vol/new_asm/LUN0 5
vserver1   800DT$HuVWBX /vol/new_asm/LUN1 6
vserver1   800DT$HuVWBZ /vol/new_asm/LUN2 6
vserver1   800DT$HuVWBZ /vol/new_asm/LUN3 5
vserver1   800DT$HuVWBa /vol/new_asm/LUN5 4
vserver1   800DT$HuVWBb /vol/new_asm/LUN6 4
vserver1   800DT$HuVWBc /vol/new_asm/LUN7 4
vserver1   800DT$HuVWBd /vol/new_asm/LUN8 4
vserver1   800DT$HuVWBe /vol/new_asm/LUN9 4
vserver1   800DT$HuVWBf /vol/new_lvm/LUN0 5
vserver1   800DT$HuVWBg /vol/new_lvm/LUN1 4
vserver1   800DT$HuVWBh /vol/new_lvm/LUN2 4
vserver1   800DT$HuVWBh /vol/new_lvm/LUN3 3
vserver1   800DT$HuVWBj /vol/new_lvm/LUN4 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN6 4
vserver1   800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1   800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1   800DT$HuVWBn /vol/new_lvm/LUN9 2
20 entries were displayed.
```

オフラインプロセスが必要な場合は、コマンドがすべての移行が正常に完了したことを示すまで、サービスの再検出または再開を遅らせて `lun import show` ください。その後、の説明に従って移行プロセスを完了できます **"Foreign LUN Import—完了"**。



オンライン移行が必要な場合は、新しいホーム内のLUNの再検出に進み、サービスを起動します。

#### SCSIデバイスの変更をスキャン

ほとんどの場合、新しいLUNを再検出する最も簡単なオプションは、ホストを再起動することです。これにより、古いデバイスが自動的に削除され、新しいLUNがすべて適切に検出され、マルチパスデバイスなどの関連デバイスが構築されます。この例では、デモ用の完全オンラインプロセスを示しています。

注意：ホストを再起動する前に、`/etc/fstab` 移行されたSANリソースについては、コメントアウトされています。これを行わず、LUNアクセスに問題があると、OSがブートしない可能性があります。この状況ではデータが破損することはありません。ただし、レスキューモードまたは同様のモードで起動し、`/etc/fstab` これにより、OSを起動してトラブルシューティングを有効にすることができます。

この例で使用しているLinuxバージョンのLUNは、`rescan-scsi-bus.sh` コマンドを実行しますコマンドが成功すると、各LUNパスが出力に表示されます。出力は解釈が難しい場合がありますが、ゾーニングとigroupの設定が正しい場合は、`NETAPP` ベンダー文字列。

```

[root@host1 /]# rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
Scanning host 0 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
  Scanning for device 0 2 0 0 ...
OLD: Host: scsi0 Channel: 02 Id: 00 Lun: 00
      Vendor: LSI          Model: RAID SAS 6G 0/1  Rev: 2.13
      Type:   Direct-Access                      ANSI SCSI revision: 05
Scanning host 1 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
  Scanning for device 1 0 0 0 ...
OLD: Host: scsi1 Channel: 00 Id: 00 Lun: 00
      Vendor: Optiarc      Model: DVD RW AD-7760H  Rev: 1.41
      Type:   CD-ROM                      ANSI SCSI revision: 05
Scanning host 2 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 3 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 4 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 5 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 6 for SCSI target IDs 0 1 2 3 4 5 6 7, all LUNs
Scanning host 7 for all SCSI target IDs, all LUNs
  Scanning for device 7 0 0 10 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 10
      Vendor: NETAPP      Model: LUN C-Mode          Rev: 8300
      Type:   Direct-Access                      ANSI SCSI revision: 05
  Scanning for device 7 0 0 11 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 11
      Vendor: NETAPP      Model: LUN C-Mode          Rev: 8300
      Type:   Direct-Access                      ANSI SCSI revision: 05
  Scanning for device 7 0 0 12 ...
...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 18
      Vendor: NETAPP      Model: LUN C-Mode          Rev: 8300
      Type:   Direct-Access                      ANSI SCSI revision: 05
  Scanning for device 9 0 1 19 ...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 19
      Vendor: NETAPP      Model: LUN C-Mode          Rev: 8300
      Type:   Direct-Access                      ANSI SCSI revision: 05
0 new or changed device(s) found.
0 remapped or resized device(s) found.
0 device(s) removed.

```

#### マルチハステハイスノカクニン

LUN検出プロセスではマルチパスデバイスの再作成もトリガーされますが、Linuxのマルチパスドライバでは時折問題が発生することがわかっています。の出力 `multipath - ll` 出力が想定どおりに表示されることを確認する必要があります。たとえば、次の出力は、に関連付けられているマルチパスデバイスを示しています。NETAPP ベンダー文字列。各デバイスには4つのパスがあり、2つはプライオリティ50、2つはプライオリティ10です。正確な出力はLinuxのバージョンによって異なりますが、この出力は想定どおりです。



使用するLinuxのバージョンに対応するHost Utilitiesのマニュアルを参照して、  
/etc/multipath.conf 設定が正しい。

```
[root@host1 /]# multipath -ll
3600a098038303558735d493762504b36 dm-5 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:4 sdat 66:208 active ready running
| `-- 9:0:1:4 sdbn 68:16 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
   |- 7:0:0:4 sdf 8:80 active ready running
   `-- 9:0:0:4 sdz 65:144 active ready running
3600a098038303558735d493762504b2d dm-10 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:8 sdax 67:16 active ready running
| `-- 9:0:1:8 sdbx 68:80 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
   |- 7:0:0:8 sdj 8:144 active ready running
   `-- 9:0:0:8 sdad 65:208 active ready running
...
3600a098038303558735d493762504b37 dm-8 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:5 sdau 66:224 active ready running
| `-- 9:0:1:5 sdbo 68:32 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
   |- 7:0:0:5 sdg 8:96 active ready running
   `-- 9:0:0:5 sdaa 65:160 active ready running
3600a098038303558735d493762504b4b dm-22 NETAPP ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:19 sdbi 67:192 active ready running
| `-- 9:0:1:19 sdcc 69:0 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
   |- 7:0:0:19 sdu 65:64 active ready running
   `-- 9:0:0:19 sdao 66:128 active ready running
```

#### LVMボリュームグループの再アクティブ化

LVM LUNが正しく検出されていれば、vgchange --activate y コマンドは成功するはずです。これは、

論理ボリュームマネージャの価値を示す良い例です。ボリュームグループのメタデータはLUN自体に書き込まれるため、LUNのWWNやシリアル番号の変更は重要ではありません。

OSがLUNをスキャンし、LUNに書き込まれている少量のデータが検出され、LUNがLUNに属する物理ボリュームであることがわかりました。 `sanvg volumegroup`。その後、必要なすべてのデバイスを構築しました。必要なのは、ボリュームグループを再アクティブ化することだけです。

```
[root@host1 ~]# vgchange --activate y sanvg
Found duplicate PV fpCzdLTuKfy2xDZjailNliJh3TjLUBiT: using
/dev/mapper/3600a098038303558735d493762504b46 not /dev/sdp
Using duplicate PV /dev/mapper/3600a098038303558735d493762504b46 from
subsystem DM, ignoring /dev/sdp
2 logical volume(s) in volume group "sanvg" now active
```

#### ファイルシステムの再マウント

ボリューム・グループを再アクティブ化すると、元のデータをすべてそのまま使用してファイル・システムをマウントできます。前述したように、バックグループでデータレプリケーションがまだアクティブであっても、ファイルシステムは完全に動作します。

```
[root@host1 ~]# mount /orabin
[root@host1 ~]# mount /backups
[root@host1 ~]# df -k
```

Filesystem	1K-blocks	Used	Available	Use%
Mounted on				
/dev/mapper/rhel-root	52403200	8837100	43566100	17% /
devtmpfs	65882776	0	65882776	0% /dev
tmpfs	6291456	84	6291372	1%
/dev/shm				
tmpfs	65898668	9884	65888784	1% /run
tmpfs	65898668	0	65898668	0%
/sys/fs/cgroup				
/dev/sda1	505580	224828	280752	45% /boot
fas8060-nfs-public:/install	199229440	119368256	79861184	60%
/install				
fas8040-nfs-routable:/snapomatic	9961472	30528	9930944	1%
/snapomatic				
tmpfs	13179736	16	13179720	1%
/run/user/42				
tmpfs	13179736	0	13179736	0%
/run/user/0				
/dev/mapper/sanvg-lvorabin	20961280	12357456	8603824	59%
/orabin				
/dev/mapper/sanvg-lvbackups	73364480	62947536	10416944	86%
/backups				

## ASMデハイスノサイズキャン

ASMLibデバイスは、SCSIデバイスが再スキャンされたときに再検出されているはずですが。再検出をオンラインで確認するには、ASMLibを再起動してからディスクをスキャンします。



この手順は、ASMLibを使用するASM構成にのみ関連します。

注意：ASMLibを使用しない場合は、`/dev/mapper` デバイスは自動的に再作成されているはずですが。ただし、権限が正しくない可能性があります。ASMLibがない場合は、ASMの基盤となるデバイスに特別な権限を設定する必要があります。これは通常、次のいずれかの特別なエントリによって達成されます。

`/etc/multipath.conf` または `udev` ルール、または両方のルールセットに含まれている可能性があります。ASMデバイスに正しいアクセス許可が設定されていることを確認するには、WWNまたはシリアル番号に関する環境の変更を反映するために、これらのファイルの更新が必要になる場合があります。

この例では、ASMLibを再起動してディスクをスキャンすると、元の環境と同じ10個のASM LUNが表示されます。

```
[root@host1 /]# oracleasm exit
Unmounting ASMLib driver filesystem: /dev/oracleasm
Unloading module "oracleasm": oracleasm
[root@host1 /]# oracleasm init
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMLib driver filesystem: /dev/oracleasm
[root@host1 /]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "ASM0"
Instantiating disk "ASM1"
Instantiating disk "ASM2"
Instantiating disk "ASM3"
Instantiating disk "ASM4"
Instantiating disk "ASM5"
Instantiating disk "ASM6"
Instantiating disk "ASM7"
Instantiating disk "ASM8"
Instantiating disk "ASM9"
```

## グリッドサービスの再起動

LVMデバイスとASMデバイスがオンラインで使用可能になったので、グリッドサービスを再起動できます。

```
[root@host1 /]# cd /orabin/product/12.1.0/grid/bin
[root@host1 bin]# ./crsctl start has
```

## データベースの再起動

グリッドサービスが再起動されたら、データベースを起動できます。ASMサービスが完全に使用可能になるまで数分待ってからデータベースを起動しなければならない場合があります。

```
[root@host1 bin]# su - oracle
[oracle@host1 ~]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base has been set to /orabin
[oracle@host1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to an idle instance.
SQL> startup
ORACLE instance started.
Total System Global Area 3221225472 bytes
Fixed Size 4502416 bytes
Variable Size 1207962736 bytes
Database Buffers 1996488704 bytes
Redo Buffers 12271616 bytes
Database mounted.
Database opened.
SQL>
```

完了

ホスト側から見ると移行は完了しますが、インポート関係が削除されるまでは外部アレイからI/Oが提供されます。

関係を削除する前に、すべてのLUNの移行プロセスが完了していることを確認する必要があります。

```
Cluster01::*> lun import show -vserver vserver1 -fields foreign-
disk,path,operational-state
vserver    foreign-disk path                                operational-state
-----
vserver1 800DT$HuVWB/ /vol/new_asm/LUN4 completed
vserver1 800DT$HuVWBW /vol/new_asm/LUN0 completed
vserver1 800DT$HuVWBX /vol/new_asm/LUN1 completed
vserver1 800DT$HuVWBZ /vol/new_asm/LUN2 completed
vserver1 800DT$HuVWBa /vol/new_asm/LUN3 completed
vserver1 800DT$HuVWBb /vol/new_asm/LUN5 completed
vserver1 800DT$HuVWBc /vol/new_asm/LUN6 completed
vserver1 800DT$HuVWBd /vol/new_asm/LUN7 completed
vserver1 800DT$HuVWBd /vol/new_asm/LUN8 completed
vserver1 800DT$HuVWBe /vol/new_asm/LUN9 completed
vserver1 800DT$HuVWBf /vol/new_lvm/LUN0 completed
vserver1 800DT$HuVWBg /vol/new_lvm/LUN1 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN9 completed
20 entries were displayed.
```

インポート関係を削除します

移行プロセスが完了したら、移行関係を削除します。I/O処理が完了すると、ONTAP上のドライブからのみI/Oが提供されます。

```
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN9
```

外部LUNの登録解除

最後に、ディスクを変更して is-foreign 指定。

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign false
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign false
Cluster01::*>
```

## プロトコル変更

LUNへのアクセスに使用するプロトコルの変更は、一般的な要件です。

場合によっては、全体的な戦略の一環としてデータをクラウドに移行することもあります。TCP/IPはクラウドのプロトコルであり、FCからiSCSIに変更することで、さまざまなクラウド環境への移行が容易になります。また、IP SANのコスト削減を活用するためにiSCSIが望ましい場合もあります。移行では、一時的な手段として別のプロトコルが使用されることがあります。たとえば、外部アレイとONTAPベースのLUNを同じHBA上に共存させることができない場合は、iSCSI LUNを使用して古いアレイからデータをコピーできます。その後、古いLUNをシステムから削除したあとにFCに変換し直すことができます。

次の手順はFCからiSCSIへの変換を示していますが、全体的な原則はiSCSIからFCへの逆変換に適用されます。

### iSCSIイニシエータのインストール

ほとんどのオペレーティングシステムには、デフォルトでソフトウェアiSCSIイニシエータが含まれていますが、含まれていない場合は簡単にインストールできます。

```
[root@host1 /]# yum install -y iscsi-initiator-utils
Loaded plugins: langpacks, product-id, search-disabled-repos,
subscription-
               : manager
Resolving Dependencies
--> Running transaction check
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.el7 will be
updated
--> Processing Dependency: iscsi-initiator-utils = 6.2.0.873-32.el7 for
package: iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7 will be
an update
--> Running transaction check
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.el7 will
be updated
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
```



```

will be an update
--> Finished Dependency Resolution
Dependencies Resolved

=====
===
Package                                Arch    Version                                Repository
Size
=====
===
Updating:
  iscsi-initiator-utils                x86_64  6.2.0.873-32.0.2.el7                 ol7_latest 416
k
Updating for dependencies:
  iscsi-initiator-utils-iscsiuio       x86_64  6.2.0.873-32.0.2.el7                 ol7_latest  84
k
Transaction Summary
=====
===
Upgrade 1 Package (+1 Dependent package)
Total download size: 501 k
Downloading packages:
No Presto metadata available for ol7_latest
(1/2): iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_6 | 416 kB    00:00
(2/2): iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2. |  84 kB    00:00
-----
---
Total                                2.8 MB/s | 501 kB
00:00Cluster01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating    : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
1/4
  Updating    : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
2/4
  Cleanup     : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
  Cleanup     : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
rhel-7-server-eus-rpms/7Server/x86_64/productid | 1.7 kB    00:00
rhel-7-server-rpms/7Server/x86_64/productid    | 1.7 kB    00:00
  Verifying   : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
1/4
  Verifying   : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
2/4

```

```
Verifying   : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
Verifying   : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
Updated:
  iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7
Dependency Updated:
  iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
Complete!
[root@host1 /]#
```

## iSCSIイニシエータ名の識別

インストールプロセス中に一意のiSCSIイニシエータ名が生成されます。Linuxの場合は、`/etc/iscsi/initiatorname.iscsi` ファイル。この名前は、IP SAN上のホストを識別するために使用されます。

```
[root@host1 /]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1992-05.com.redhat:497bd66ca0
```

## 新しいイニシエータグループを作成する

イニシエータグループ (igroup) は、ONTAP LUNマスキングアーキテクチャの一部です。新しく作成したLUNには、ホストに最初にアクセスを許可しないかぎりアクセスできません。そのためには、アクセスが必要なFC WWNまたはiSCSIイニシエータ名のいずれかをリストするigroupを作成します。

この例では、LinuxホストのiSCSIイニシエータを含むigroupを作成しています。

```
Cluster01::*> igroup create -igroup linuxiscsi -protocol iscsi -ostype
linux -initiator iqn.1994-05.com.redhat:497bd66ca0
```

## 環境をシャットダウンする

LUNプロトコルを変更する前に、LUNを完全に休止する必要があります。変換するLUNのいずれかのデータベースをシャットダウンし、ファイルシステムをディスマウントし、ボリュームグループを非アクティブ化する必要があります。ASMを使用する場合は、ASMディスクグループがディスマウントされていることを確認し、すべてのグリッドサービスをシャットダウンします。

## FCネットワークからのLUNのマッピング解除

LUNが完全に休止されたら、元のFC igroupからマッピングを削除します。

```
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN0 -igroup linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN1 -igroup linuxhost
...
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup linuxhost
```

### IPネットワークへのLUNの再マッピング

新しいiSCSIベースのイニシエータグループに各LUNへのアクセスを許可します。

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup linuxiscsi
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup linuxiscsi
Cluster01::*>
```

### iSCSIターゲットの検出

iSCSI検出には2つのフェーズがあります。1つ目はターゲットの検出です。これは、LUNの検出とは異なります。。iscsiadm 次のコマンドは、-p argument およびには、iSCSIサービスを提供するすべてのIPアドレスとポートのリストが格納されます。この場合、デフォルトポート3260にiSCSIサービスを持つIPアドレスが4つあります。



いずれかのターゲットIPアドレスに到達できない場合、このコマンドは完了までに数分かかることがあります。

```
[root@host1 ~]# iscsiadm -m discovery -t st -p fas8060-iscsi-public1
10.63.147.197:3260,1033 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
10.63.147.198:3260,1034 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.203:3260,1030 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.202:3260,1029 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
```

## iSCSI LUNの検出

iSCSIターゲットが検出されたら、iSCSIサービスを再起動して使用可能なiSCSI LUNを検出し、マルチパスやASMLibデバイスなどの関連デバイスを構築します。

```
[root@host1 ~]# service iscsi restart
Redirecting to /bin/systemctl restart iscsi.service
```

## 環境の再起動

ボリュームグループの再アクティブ化、ファイルシステムの再マウント、RACサービスの再起動などを実行して、環境を再起動します。予防措置としてNetApp、変換プロセスの完了後にサーバを再起動して、すべての構成ファイルが正しいことと古いデバイスがすべて削除されることを確認することをお勧めします。

注意：ホストを再起動する前に、`/etc/fstab` 移行されたSANリソースについては、コメントアウトされています。この手順を実行せず、LUNアクセスに問題があると、OSがブートしない可能性があります。この問題はデータに損傷を与えません。ただし、レスキューモードまたは同様のモードで起動して修正するのは非常に不便な場合があります。`/etc/fstab` OSを起動してトラブルシューティング作業を開始できるようにします。

# サンプルスクリプト

ここで紹介するスクリプトは、さまざまなOSおよびデータベースタスクのスクリプト作成方法の例として提供されています。それらはそのまま供給されます。特定の手順のサポートが必要な場合は、NetAppまたはNetAppリセラーにお問い合わせください。

## データベースのシャットダウン

次のPerlスクリプトは、Oracle SIDの引数を1つ指定してデータベースをシャットダウンします。Oracleユーザまたはrootとして実行できます。

```

#!/usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
77 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
';}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF4
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
`;};
print @out;
if ("@out" =~ /ORACLE instance shut down/) {
print "$oraclesid shut down\n";
exit 0;}
elsif ("@out" =~ /Connected to an idle instance/) {
print "$oraclesid already shut down\n";
exit 0;}
else {
print "$oraclesid failed to shut down\n";
exit 1;}

```

## データベースの起動

次のPerlスクリプトは、Oracle SIDの引数を1つ指定してデータベースをシャットダウンします。Oracleユーザまたはrootとして実行できます。

```

#!/usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
`;
}
else {
@out=`. oraenv << EOF3
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
`;
};
print @out;
if ("@out" =~ /Database opened/) {
print "$oraclesid started\n";
exit 0;}
elsif ("@out" =~ /cannot start already-running ORACLE/) {
print "$oraclesid already started\n";
exit 1;}
else {
78 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
print "$oraclesid failed to start\n";
exit 1;}

```

## ファイルシステムを読み取り専用に変換

次のスクリプトは、ファイルシステム引数を取り、ディスマウントして読み取り専用として再マウントしようとしています。これは、データをレプリケートするためにファイルシステムの可用性を維持しつつ、偶発的な破損から保護する必要がある移行プロセスで役立ちます。

```

#!/usr/bin/perl
use strict;
#use warnings;
my $filesystem=$ARGV[0];
my @out=`umount '$filesystem'`;
if ($? == 0) {
    print "$filesystem unmounted\n";
    @out = `mount -o ro '$filesystem'`;
    if ($? == 0) {
        print "$filesystem mounted read-only\n";
        exit 0;}}
else {
    print "Unable to unmount $filesystem\n";
    exit 1;}
print @out;

```

## ファイルシステムの交換

次のスクリプト例は、あるファイルシステムを別のファイルシステムに置き換えるために使用します。`/etc/fstab` ファイルを編集するので、rootとして実行する必要があります。古いファイルシステムと新しいファイルシステムの単一のカンマ区切り引数を受け入れます。

1. ファイルシステムを交換するには、次のスクリプトを実行します。

```

#!/usr/bin/perl
use strict;
#use warnings;
my $oldfs;
my $newfs;
my @oldfstab;
my @newfstab;
my $source;
my $mountpoint;
my $leftover;
my $oldfstabentry='';
my $newfstabentry='';
my $migratedfstabentry='';
($oldfs, $newfs) = split ('', $ARGV[0]);
open(my $filehandle, '<', '/etc/fstab') or die "Could not open
/etc/fstab\n";
while (my $line = <$filehandle>) {
    chomp $line;
    ($source, $mountpoint, $leftover) = split(/[, ]/, $line, 3);
    if ($mountpoint eq $oldfs) {
        $oldfstabentry = "#Removed by swap script $source $oldfs $leftover";}

```

```

elif ($mountpoint eq $newfs) {
    $newfstabentry = "#Removed by swap script $source $newfs $leftover";
    $migratedfstabentry = "$source $oldfs $leftover";}
else {
    push (@newfstab, "$line\n")}}
79 Migration of Oracle Databases to NetApp Storage Systems © 2021
NetApp, Inc. All rights reserved
push (@newfstab, "$oldfstabentry\n");
push (@newfstab, "$newfstabentry\n");
push (@newfstab, "$migratedfstabentry\n");
close($filehandle);
if ($oldfstabentry eq ''){
    die "Could not find $oldfs in /etc/fstab\n";}
if ($newfstabentry eq ''){
    die "Could not find $newfs in /etc/fstab\n";}
my @out=`umount '$newfs'`;
if ($? == 0) {
    print "$newfs unmounted\n";}
else {
    print "Unable to unmount $newfs\n";
    exit 1;}
@out=`umount '$oldfs'`;
if ($? == 0) {
    print "$oldfs unmounted\n";}
else {
    print "Unable to unmount $oldfs\n";
    exit 1;}
system("cp /etc/fstab /etc/fstab.bak");
open ($filehandle, ">", '/etc/fstab') or die "Could not open /etc/fstab
for writing\n";
for my $line (@newfstab) {
    print $filehandle $line;}
close($filehandle);
@out=`mount '$oldfs'`;
if ($? == 0) {
    print "Mounted updated $oldfs\n";
    exit 0;}
else{
    print "Unable to mount updated $oldfs\n";
    exit 1;}
exit 0;

```

このスクリプトの使用例として、/oradata の移行先 /neworadata および /logs の移行先 /newlogs。このタスクを実行する最も簡単な方法の1つは、単純なファイルコピー操作を使用して、新しいデバイスを元のマウントポイントに再配置することです。



2. 古いファイルシステムと新しいファイルシステムが /etc/fstab ファイルは次のとおりです。

```
cluster01:/vol_oradata /oradata nfs rw,bg,vers=3,rsiz=65536,wsiz=65536
0 0
cluster01:/vol_logs /logs nfs rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
cluster01:/vol_newlogs /newlogs nfs rw,bg,vers=3,rsiz=65536,wsiz=65536
0 0
```

3. このスクリプトを実行すると、現在のファイルシステムがアンマウントされ、新しいファイルシステムに置き換えられます。

```
[root@jpsc3 scripts]# ./swap.fs.pl /oradata,/neworadata
/neworadata unmounted
/oradata unmounted
Mounted updated /oradata
[root@jpsc3 scripts]# ./swap.fs.pl /logs,/newlogs
/newlogs unmounted
/logs unmounted
Mounted updated /logs
```

4. このスクリプトでは、/etc/fstab 必要に応じてファイルを作成この例では、次の変更が含まれています。

```
#Removed by swap script cluster01:/vol_oradata /oradata nfs
rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
cluster01:/vol_neworadata /oradata nfs
rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_logs /logs nfs
rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_newlogs /newlogs nfs
rw,bg,vers=3,rsiz=65536,wsiz=65536 0 0
cluster01:/vol_newlogs /logs nfs rw,bg,vers=3,rsiz=65536,wsiz=65536 0
0
```

## データベース移行の自動化

この例では、シャットダウン、起動、およびファイルシステム置換スクリプトを使用して移行を完全に自動化する方法を示します。

```

#!/usr/bin/perl
use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my @oldfs;
my @newfs;
my $x=1;
while ($x < scalar(@ARGV)) {
    ($oldfs[$x-1], $newfs[$x-1]) = split (',', $ARGV[$x]);
    $x+=1;}
my @out=`./dbshut.pl '$oraclesid'`;
print @out;
if ($? ne 0) {
    print "Failed to shut down database\n";
    exit 0;}
$x=0;
while ($x < scalar(@oldfs)) {
    my @out=`./mk.fs.readonly.pl '$oldfs[$x]'`;
    if ($? ne 0) {
        print "Failed to make filesystem $oldfs[$x] readonly\n";
        exit 0;}
    $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
    my @out=`rsync -rlpogt --stats --progress --exclude='.snapshot'
'$oldfs[$x]/' '$newfs[$x]/'`;
    print @out;
    if ($? ne 0) {
        print "Failed to copy filesystem $oldfs[$x] to $newfs[$x]\n";
        exit 0;}
    else {
        print "Succesfully replicated filesystem $oldfs[$x] to
$newfs[$x]\n";}
    $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
    print "swap $x $oldfs[$x] $newfs[$x]\n";
    my @out=`./swap.fs.pl '$oldfs[$x],$newfs[$x]'`;
    print @out;
    if ($? ne 0) {
        print "Failed to swap filesystem $oldfs[$x] for $newfs[$x]\n";
        exit 1;}
    else {
        print "Swapped filesystem $oldfs[$x] for $newfs[$x]\n";}
    $x+=1;}
my @out=`./dbstart.pl '$oraclesid'`;

```

```
print @out;
```

## ファイルの場所を表示する

このスクリプトは、多数の重要なデータベースパラメータを収集し、読みやすい形式で出力します。このスクリプトは、データレイアウトを確認する場合に役立ちます。また、他の用途に合わせてスクリプトを変更することもできます。

```
#!/usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
    my $command = $_[0];
    my @lines;
    my $uid=$<;
    if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORAACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
        `; }
    else {
        $command=~s/\\\\\\\\\\\\\\\\/\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
        `; };
    return @lines;
}
print "\n";
@out=dosql('select name from v\\\\\\\\\\\\$datafile;');
print "$oraclesid datafiles:\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
```

```

print "\n";
@out=dosql('select member from v\\\\\\$logfile;');
print "$oraclesid redo logs:\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name from v\\\\\\$tempfile;');
print "$oraclesid temp datafiles:\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('show parameter spfile;');
print "$oraclesid spfile\n";
for $line (@out) {
    chomp($line);
    if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name||'\ '||value from v\\\\\\$parameter where
isdefault=\'FALSE\';');
print "$oraclesid key parameters\n";
for $line (@out) {
    chomp($line);
    if ($line =~ /control_files/) {print "$line\n";}
    if ($line =~ /db_create/) {print "$line\n";}
    if ($line =~ /db_file_name_convert/) {print "$line\n";}
    if ($line =~ /log_archive_dest/) {print "$line\n";}}
    if ($line =~ /log_file_name_convert/) {print "$line\n";}
    if ($line =~ /pdb_file_name_convert/) {print "$line\n";}
    if ($line =~ /spfile/) {print "$line\n";}
print "\n";

```

## ASM移行のクリーンアップ

```

#!/usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
    my $command = @_ [0];
    my @lines;
    my $uid=$<;

```

```

        if ($uid == 0) {
            @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"

            `; }
            else {
                $command=~s/\\\\\\\\\\\\\\\\/\\\\/g;
                @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2

                `; }
            return @lines}
            print "\n";
            @out=dosql('select name from v\\\\\\\\\\\\$datafile;');
            print @out;
            print "shutdown immediate;\n";
            print "startup mount;\n";
            print "\n";
            for $line (@out) {
                if (length($line) > 1) {
                    chomp($line);
                    ($first, $second,$third,$fourth)=split('_', $line);
                    $fourth =~ s/^TS-//;
                    $newname=lc("$fourth.dbf");
                    $path2file=$line;
                    $path2file=~ /(^.*\\.\\/)/;
                    print "host mv $line $1$newname\n";}}
            print "\n";
            for $line (@out) {
                if (length($line) > 1) {
                    chomp($line);
                    ($first, $second,$third,$fourth)=split('_', $line);
                    $fourth =~ s/^TS-//;
                    $newname=lc("$fourth.dbf");
                    $path2file=$line;
                    $path2file=~ /(^.*\\.\\/)/;
                    print "alter database rename file '$line' to

```

```
'$1$newname';\n";}}  
print "alter database open;\n";  
print "\n";
```

## ASMからファイルシステム名への変換

```

set serveroutput on;
set wrap off;
declare
    cursor df is select file#, name from v$datafile;
    cursor tf is select file#, name from v$tempfile;
    cursor lf is select member from v$logfile;
    firstline boolean := true;
begin
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('Parameters for log file conversion:');
    dbms_output.put_line(CHR(13));
    dbms_output.put('*.log_file_name_convert = ');
    for lfrec in lf loop
        if (firstline = true) then
            dbms_output.put('''' || lfrec.member || ''', ');
            dbms_output.put(''''/NEW_PATH/' ||
regex_replace(lfrec.member, '^.*./', '') || ''');
        else
            dbms_output.put(', ''' || lfrec.member || ''', ');
            dbms_output.put(''''/NEW_PATH/' ||
regex_replace(lfrec.member, '^.*./', '') || ''');
        end if;
        firstline:=false;
    end loop;
    dbms_output.put_line(CHR(13));
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('rman duplication script:');
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('run');
    dbms_output.put_line('{');
    for dfrec in df loop
        dbms_output.put_line('set newname for datafile ' ||
dfrec.file# || ' to ''' || dfrec.name || ''';');
    end loop;
    for tfrec in tf loop
        dbms_output.put_line('set newname for tempfile ' ||
tfrec.file# || ' to ''' || tfrec.name || ''';');
    end loop;
    dbms_output.put_line('duplicate target database for standby backup
location INSERT_PATH_HERE;');
    dbms_output.put_line('}');
end;
/

```

## データベースでログを再生

このスクリプトは、マウントモードのデータベースに対してOracle SIDの引数を1つ指定し、現在使用可能なすべてのアーカイブログを再生します。

```
#!/usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
84 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
`;
}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
`;
}
print @out;
```

## スタンバイデータベースでログを再生

このスクリプトは、スタンバイデータベース用に設計されている点を除き、上記のスクリプトと同じです。



```

#!/usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
';}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
';}
}
print @out;

```

## 著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータ ソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。