テクニカル レポート

Oracleのデータ保護：バックアップ、リカバリ、レプリケーション、ディザスタリカバリ

NetApp
Jeffrey Steiner
2021年4月| TR-4591

概要

本レポートに指定された環境、構成、バージョンがお客様の環境に対応しているかどうかについては、Interoperability Matrix Tool（IMT）を参照してください。

<<本レポートは機械翻訳による参考訳です。公式な内容はオリジナルである英語版をご確認ください。>>

目次

表一覧

図一覧

ONTAPによるデータベースデータ保護

最もミッションクリティカルなデータはデータベースにあります。企業はデータへのアクセスなしでは業務を遂行できず、場合によってはデータによってビジネスが決まることもあります。このようなデータは保護する必要がありますが、データ保護では、使用可能なバックアップを確保するだけでなく、バックアップを安全に保管するだけでなく、迅速かつ確実に実行することも重要です。データ保護のもう1つの側面は、データリカバリです。データにアクセスできなくなると企業は影響を受け、データがリストアされるまで操作できなくなる可能性があります。このプロセスは高速で信頼性が必要です。最後に、ほとんどのデータベースを災害から保護する必要があります。つまり、データベースのレプリカを維持する必要があります。レプリカは十分に最新である必要があります。また'レプリカを完全に動作可能なデータベースにするには'迅速かつ簡単に行う必要があります

データベースのデータ保護

データベースのデータ保護アーキテクチャは、ビジネス要件によって定義する必要があります。これらの要件には、リカバリの速度、許容される最大データ損失、バックアップの保持ニーズなどの要因が含まれます。データ保護計画では、データの保持とリストアに関するさまざまな規制要件も考慮する必要があります。最後に、データベースのデータ保護に関するさまざまなデータリカバリシナリオを評価することを忘れないでください。考慮すべき状況としては、ユーザやアプリケーションのエラーに起因する一般的で予測可能なリカバリや、サイトの完全な停止によって発生する可能性のあるディザスタリカバリ作業などがあります。

データ保護ポリシーとリカバリポリシーのわずかな変更は、ストレージ、バックアップ、リカバリのアーキテクチャ全体に大きな影響を与える可能性があります。データ保護アーキテクチャが複雑にならないように、設計作業を開始する前に標準を定義して文書化することが重要です。不要な機能や保護レベルは、不要なコストや管理オーバーヘッドにつながります。当初見過ごされていた要件も、プロジェクトを間違った方向に進めたり、最終的な設計変更が必要になる可能性があります。

目標復旧時間

Recovery Time Objective（RTO；目標復旧時間）は、サービスのリカバリに許容される最大時間を定義します。たとえば、人事データベースのRTOが24時間になる可能性があります。これは、営業日中にこのデータにアクセスできなくなるのは不便ですが、ビジネスを継続できるためです。一方、銀行の総勘定元帳をサポートするデータベースでは、数分または数秒でRTOを測定できます。実際のサービス停止とネットワークパケットの損失などの日常的なイベントを区別する方法が必要なため、RTOをゼロにすることはできません。ただし、一般的な要件はRTOがほぼゼロです。

目標復旧時点

Recovery Point Objective（RPO；目標復旧時点）は、最大許容データ損失を定義します。データベースのコンテキストでは、通常、RPOは、特定の状況で失われる可能性のあるログデータの量です。製品のバグやユーザエラーによってデータベースが破損した一般的なリカバリシナリオでは、RPOはゼロ（データ損失がないことを意味します）にする必要があります。リカバリ手順では、データベースファイルの以前のコピーをリストアし、ログファイルを再生して、データベースを希望する時点の状態にします。この処理に必要なログファイルは元の場所にすでに存在している必要があります。

通常とは異なる状況では、ログデータが失われる可能性があります。たとえば、偶発的または悪意のある `rm -rf *` データベースファイルがあると、すべてのデータが削除される可能性があります。唯一の選択肢は、ログファイルを含むバックアップからリストアすることですが、一部のデータは必然的に失われます。従来のバックアップ環境でRPOを向上させる唯一の方法は、ログデータのバックアップを繰り返し実行することです。ただし、このオプションには、継続的なデータ移動と、バックアップシステムを継続的に実行されるサービスとして維持することが困難なため、制限があります。高度なストレージシステムのメリットの1つは、偶発的または悪意のあるファイルの破損からデータを保護し、データを移動せずにRPOを向上できることです。

ディザスタ リカバリ

ディザスタリカバリには、物理的な災害が発生した場合にサービスをリカバリするために必要なITアーキテクチャ、ポリシー、および手順が含まれます。そのようなイベントには、洪水、火災、または悪意または過失の意図を持って行動する人が含まれます。

ディザスタリカバリは、単なるリカバリ手順ではありません。これは、さまざまなリスクを特定し、データリカバリとサービス継続性の要件を定義し、適切なアーキテクチャと関連手順を提供する完全なプロセスです。

データ保護の要件を確立するには、一般的なRPOとRTOの要件と、ディザスタリカバリに必要なRPOとRTOの要件を区別することが重要です。一部のデータベース環境では、比較的通常のユーザエラーや火災によってデータセンターが破壊されたことが原因でデータ損失が発生した場合に、RPOをゼロ、RTOをほぼゼロにする必要があります。ただし、これらの高レベルの保護にはコストと管理上の影響があります。

一般に、ディザスタ以外のデータリカバリ要件は、次の2つの理由で厳しくする必要があります。まず、データベースに損害を与えるアプリケーションのバグやユーザエラーは、ほとんど避けられないほど予測可能です。2つ目は、ストレージシステムが破損していないかぎり、RPOをゼロにしてRTOを短縮できるバックアップ戦略を設計することです。このようなデータリカバリ要件は予見可能であり、比較的容易に対処できるため、ローカルリカバリのRPOとRTOの目標は積極的に設定する必要があります。

ディザスタリカバリのRTOとRPOの要件はさまざまであり、災害の発生の可能性と、関連するデータ損失やビジネスの中断の結果によって大きく異なります。一般的な原則ではなく、実際のビジネスニーズに基づいて作成する必要があります。また、複数の論理的、物理的な災害シナリオも考慮する必要があります。

論理的災害

論理的災害には、ユーザによるデータの不整合、アプリケーションやOSのバグ、ソフトウェアの誤動作などがあります。論理的災害には、ウイルスやワームによる外部からの悪意のある攻撃や、アプリケーションの脆弱性を悪用した悪意のある攻撃も含まれます。この場合、物理インフラは破損していませんが、基盤となるデータは無効になります。

ランサムウェアと呼ばれる論理災害のタイプはますます一般的になりつつあり、攻撃ベクトルを使用してデータを暗号化します。暗号化はデータを損傷することはありませんが、サードパーティに支払いが行われるまで使用できなくなります。ランサムウェアのハッキングの標的にされる企業はますます増えています。

物理的災害

物理的災害には、インフラストラクチャのコンポーネントの障害がその冗長性機能を超え、データの損失やサービスの長期的な損失につながることが含まれます。たとえば、RAID保護はディスクドライブの冗長性を提供し、Host Bus Adapter（HBA；ホストバスアダプタ）を使用するとFCポートとFCケーブルの冗長性が提供されます。このようなコンポーネントのハードウェア障害は予測可能であり、可用性には影響しません。

データベース環境では、サイト全体のインフラを冗長コンポーネントで保護し、予測可能な唯一の物理的災害シナリオでサイトが完全に失われた時点まで保護できます。ディザスタリカバリ計画は、サイト間レプリケーションによって異なります。

同期および非同期のデータ保護

理想的な環境では、地理的に分散したサイト間ですべてのデータを同期的にレプリケートできます。このようなレプリケーションは、次のようないくつかの理由により、必ずしも実現可能ではありません。

- 同期レプリケーションでは、アプリケーションまたはデータベースを処理する前にすべての変更を両方の場所にレプリケートする必要があるため、書き込みレイテンシが避けられません。このようなパフォーマンスへの影響は許容できず、同期ミラーリングの使用が除外されることがよくあります。

- 100% SSDストレージの採用が増加しているため、期待されるパフォーマンスには数十万IOPSと1ミリ秒未満のレイテンシが含まれているため、書き込みレイテンシの増加に気付く可能性が高くなります。100% SSDを使用するメリットを最大限に引き出すには、ディザスタリカバリ戦略を見直す必要があります。
- データセットはバイト単位で増え続けているため、同期レプリケーションを維持するのに十分な帯域幅を確保するという課題が生じています。
- データセットも複雑化し、大規模な同期レプリケーションの管理が困難になっています。
- クラウドベースの戦略では、多くの場合、レプリケーションの距離とレイテンシが長くなり、同期ミラーリングの使用がさらに困難になります。

NetApp® は、最も厳しいデータリカバリ要件に対応する同期レプリケーションと、データベースのパフォーマンスと柔軟性を向上させる非同期ソリューションの両方を備えたソリューションを提供します。さらに、NetAppテクノロジは、Oracle DataGuardなどの多くのサードパーティのレプリケーションソリューションとシームレスに統合されます。

保持時間

データ保護戦略の最後の側面は、データの保持期間です。データの保持期間は大きく異なる場合があります。

- 一般的な要件は、プライマリサイトに夜間バックアップを14日間、セカンダリサイトにバックアップを90日間保存することです。
- 多くのお客様が'異なるメディアに保存された四半期ごとのスタンドアロンアーカイブを作成しています
- 定期的に更新されるデータベースでは、履歴データは不要であり、バックアップは数日間だけ保持する必要があります。

規制要件によっては、任意のトランザクションを365日以内にリカバリできることが求められる場合があります。

ONTAPの論理アーキテクチャ

ストレージシステムには、データを確実に保護し、データを利用できるようにするという2つの基本的な要件があります。NetApp® ONTAP® ONTAP® データ保護テクノロジの詳しい説明は、本ドキュメントの範囲外です。ただし、さまざまな障害シナリオで何が起こるかを完全に理解するには、レイヤのレビューが必要です。

データ保護

ONTAPでの論理データ保護は、次の3つの重要な要件で構成されます。

- データは不整合から保護する必要があります。
- データはディスク障害から保護する必要があります。
- データへの変更は損失から保護する必要があります。

この3つのニーズについては、以降のセクションで説明します。

ネットワークの破損:チェックサム

最も基本的なデータ保護レベルはチェックサムです。チェックサムは、データと一緒に格納される特別なエラー検出コードです。ネットワーク転送中のデータの破損は、チェックサムを使用して検出されます。場合によっては、複数のチェックサムを使用します。

たとえば、FCフレームには巡回冗長検査（CRC）と呼ばれるチェックサム形式が含まれており、転送中にペイロードが破損していないことを確認できます。送信機は、データのデータとCRCの両方を送信します。FCフレームの受信側は、受信したデータのCRCを再計算して、送信されたCRCと一致することを確認します。新しく計算されたCRCがフレームに接続されたCRCと一致しない場合、データは破損し、FCフレームは破棄または拒否されます。iSCSI I/O処理には、TCP/IPおよびイーサネットレイヤでのチェックサムが含まれます。また、保護

を強化するために、SCSIレイヤでオプションのCRC保護を含めることもできます。ワイヤ上のビットの破損はTCPレイヤまたはIPレイヤによって検出され、パケットが再送信されます。FCと同様に、SCSI CRCでエラーが発生すると、処理が破棄または拒否されます。

ドライブの破損：チェックサム

チェックサムは、ドライブに格納されているデータの整合性を検証するためにも使用されます。ドライブに書き込まれたデータブロックは、元のデータに関連付けられた予測不可能な数を生成するチェックサム機能で格納されます。ドライブからデータが読み取られると、チェックサムが再計算され、保存されているチェックサムと比較されます。一致しない場合は、データが破損しているため、RAIDレイヤでリカバリする必要があります。

データ破損：失われた書き込み

検出するのが最も困難な種類の破損の1つは、書き込みの紛失または置き忘れです。書き込みが確認応答されたら、正しい場所にあるメディアに書き込む必要があります。インプレースデータの破損は、データとともに保存されたシンプルなチェックサムを使用することで、比較的簡単に検出できます。ただし、書き込みが失われただけの場合は、以前のバージョンのデータが残っている可能性があり、チェックサムが正しいことになります。書き込みが間違った物理的な場所に配置された場合、書き込みによって他のデータが破壊されても、関連するチェックサムは保存データに対して再び有効になります。

この課題に対する解決策は次のとおりです。

- 書き込み処理には、書き込みが予想される場所を示すメタデータが含まれている必要があります。
- 書き込み処理には、何らかのバージョン識別子が含まれている必要があります。

ONTAPがブロックを書き込むときは、そのブロックが属する場所のデータも含まれます。たとえば、後続の読み取りでブロックが識別されていても、メタデータでブロックが456の場所で見つかったときに123の場所に属していることが示されている場合、書き込みは誤って配置されています。

完全に失われた書き込みを検出することは、より困難です。説明はかなり複雑ですが、基本的には、書き込み処理によってドライブ上の2つの場所が更新されるように、ONTAPはメタデータを格納します。書き込みが失われると、その後のデータおよび関連するメタデータの読み取りで、2つの異なるバージョンIDが表示されます。このあとの読み取りは、ドライブによる書き込みが完了していないことを示します。

書き込みの破損が失われたり置き忘れられたりすることは非常にまれですが、ドライブが増え続け、データセットがエクサバイト規模になると、リスクが増大します。データベースワークロードをサポートするストレージシステムには、Lost Write検出機能を含める必要があります。

ドライブ障害：RAID、RAID DP、RAID-TEC

ドライブ上のデータブロックが破損していることが検出された場合、またはドライブ全体で障害が発生して完全に使用できなくなった場合は、データを再構成する必要があります。このプロセスは、ONTAPでパリティドライブを使用して実行されます。データが複数のデータドライブにストライピングされ、パリティデータが生成されます。これは元のデータとは別に保存されます。

ONTAPは元々RAID 4を使用していました。RAID 4は、データドライブのグループごとにパリティドライブを1本使用します。その結果、グループ内のいずれかのドライブで障害が発生してもデータが失われることはありませんでした。パリティドライブで障害が発生してもデータは破損しておらず、新しいパリティドライブを構築できました。1本のデータドライブで障害が発生した場合は、残りのドライブをパリティドライブと一緒に使用して失われたデータを再生成します。

ドライブが小さい場合、2本のドライブで同時に障害が発生する可能性はほとんどありませんでした。ドライブ容量の増大に伴い、ドライブ障害発生後のデータの再構築に必要な時間も増加しています。そのため、2つ目のドライブ障害が発生してデータが失われる時間が長くなりました。また、再構築プロセスでは、稼働しているドライブに多数のI/Oが追加されます。ドライブが古くなると、負荷が増えて2つ目のドライブ障害が発生するリスクも高まります。最後に、RAID 4を継続して使用してもデータ損失のリスクが増加しなかったとしても、影響を受けるデータの量が増加するにつれて、データ損失の結果はより深刻になります。RAIDグルー

プで障害が発生した場合に失われるデータが増えるほど、データのリカバリにかかる時間が長くなり、業務の中断が長くなります。

これらの問題により、NetAppはRAID 6の一種であるNetApp RAID DP® テクノロジを開発しました。この解決策にはパリティドライブが2本含まれているため、RAIDグループ内の2本のドライブで障害が発生してもデータが失われることはありません。ドライブのサイズが大きくなるにつれて、NetAppはNetApp RAID-TEC™ テクノロジも開発し、3台目のパリティドライブを導入しました。

一部の履歴データベースのベストプラクティスでは、ストライプミラーリングとも呼ばれるRAID-10の使用を推奨しています。この機能は、2つのディスクで障害が発生するシナリオが複数あるのに対し、RAID DPでは何も発生しないため、RAID DPよりもデータ保護がさらに少なくなります。

また、パフォーマンス上の懸念から、RAID-4 / 5 / 6ではなくRAID-10を推奨するさまざまなベンダーのベストプラクティスも公開されています。これらの推奨事項は、RAIDペナルティを意味する場合があります。これらの推奨事項は正しくありますが、ONTAP内でのRAIDの実装には適用されません。パフォーマンスの問題はパリティ再生に関連しています。従来のRAID実装では、データベースによって実行されるルーチンのランダムライトを処理するには、パリティデータを再生成して書き込みを完了するために、複数のディスク読み取りが必要です。ペナルティは、書き込み処理の実行に必要な追加の読み取りIOPSとして定義されます。

書き込みはメモリでステージングされ、パリティが生成されてから単一のRAIDストライプとしてディスクに書き込まれるため、ONTAPではRAIDペナルティは発生しません。書き込み処理を完了するための読み取りは必要ありません。

要約すると、RAID DPとRAID-TECは、RAID 10と比較して使用可能な容量がはるかに多く、ドライブ障害に対する保護が強化され、パフォーマンスが低下することはありません。

ハードウェア障害からの保護:NVRAM

データベースワークロードを処理するストレージアレイでは、書き込み処理をできるだけ迅速に処理する必要があります。さらに、電源障害などの予期しないイベントから書き込み処理を損失から保護する必要があります。ハードウェア障害に対する保護とは、書き込み処理を少なくとも2つの場所に安全に格納する必要があることを意味します。

AFFシステムとFASシステムは、これらの要件を満たすためにNVRAMを利用しています。書き込みプロセスは次のように機能します。

1. インバウンド書き込みデータはRAMに格納されます。
2. ディスク上のデータに加えなければならない変更は、ローカルノードとパートナーノードの両方のNVRAMに記録されます。NVRAMは書き込みキャッシュではなく、データベースのRedoログに似たジャーナルです。通常の条件下では、読み取りは行われません。I/O処理中に電源障害が発生した場合など、リカバリにのみ使用されます。
3. その後、書き込みがホストに確認応答されます。

この段階の書き込みプロセスはアプリケーションの観点からは完了しており、データは2つの異なる場所に格納されるため、損失から保護されます。最終的に変更はディスクに書き込まれますが、書き込みが確認されたあとに実行されるためレイテンシに影響しないため、このプロセスはアプリケーションの観点からはアウトオブバンドです。このプロセスも、データベースロギングに似ています。データベースに対する変更はできるだけ早くREDOログに記録され、変更がコミットされたことが確認されます。データファイルの更新はかなり遅れて行われ、処理速度に直接影響することはありません。

コントローラで障害が発生すると、パートナーコントローラが必要なディスクの所有権を取得し、ログに記録されたデータをNVRAMに再生して、障害発生時に転送中だったI/O処理をリカバリします。

サイトおよびシェルフ障害からの保護：SyncMirrorとプレックス

NetApp SyncMirror® は、RAID DPやRAID-TECを強化するミラーリングテクノロジですが、これに代わるものではありません。2つの独立したRAIDグループの内容をミラーリングします。論理構成は次のとおりです。

- ドライブは、場所に基づいて2つのプールに構成されます。1つのプールはサイトAのすべてのドライブで構成され、2つ目のプールはサイトBのすべてのドライブで構成されます。
- 次に、アグリゲートと呼ばれる共通のストレージプールが、RAIDグループのミラーセットに基づいて作成されます。各サイトから同じ数のドライブが引き出されます。たとえば、20ドライブのSyncMirrorアグリゲートは、サイトAの10本のドライブとサイトBの10本のドライブで構成されます。
- 特定のサイトのドライブセットは、ミラーリングを使用することなく、1つ以上の完全に冗長化されたRAID-DPまたはRAID-TECグループとして自動的に構成されます。この自動構成により、サイトが停止した場合でも継続的なデータ保護が提供されます。

図1）SyncMirror

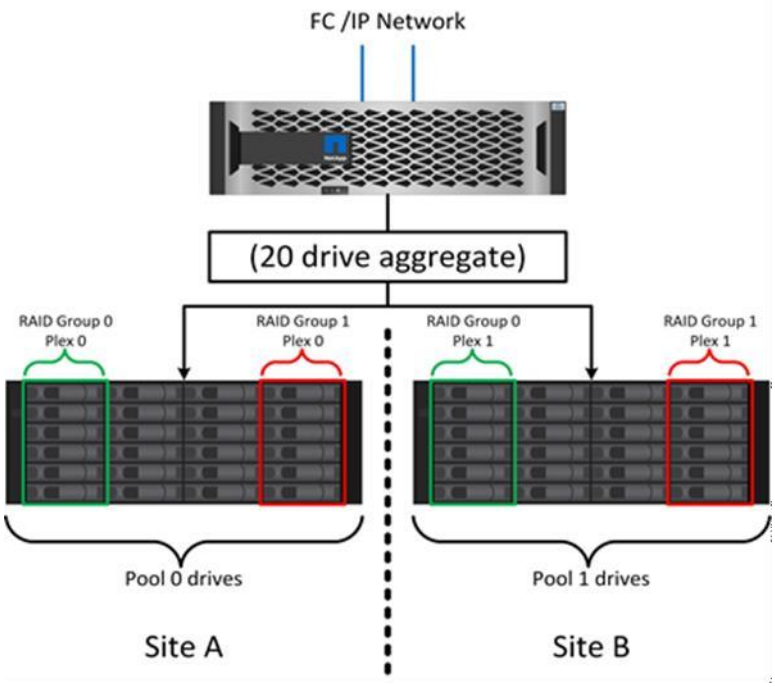


図1 は、SyncMirror構成の例を示しています。24ドライブのアグリゲートが表示されているコントローラでは、サイトAで割り当てられたシェルフの12本のドライブと、サイトBで割り当てられたシェルフの12本のドライブを使用しています。ドライブは2つのミラーRAIDグループにグループ化されました。RAIDグループ0には、サイトAの6ドライブプレックスが含まれており、サイトBの6ドライブプレックスにミラーリングされています。同様に、RAIDグループ1にはサイトAの6ドライブプレックスが含まれており、サイトBの6ドライブプレックスにミラーリングされています。

SyncMirrorは通常、NetApp® MetroClusterシステムで非同期のリモートミラーリングを提供するために使用され、各サイトにデータのコピーが1つずつ配置されます。場合によっては、1つのシステムで追加レベルの冗長性を提供するために使用されます。特に、シェルフレベルの冗長性を提供します。ドライブシェルフにはすでにデュアル電源装置とコントローラが搭載されており、全体的には板金程度ですが、追加の保護が保証される場合があります。たとえば、あるNetAppのお客様は、自動車テストで使用するモバイルリアルタイム分析プラットフォームにSyncMirrorを導入しています。システムは、独立したUPSシステムからの独立した電源供給によって供給される2つの物理ラックに分割されました。

高可用性

概要of ONTAPの高可用性機能は、本ドキュメントでは扱いません。ただし、データ保護と同様に、データベースインフラを設計する際には、この機能の基本的な理解が重要です。

HAペア

ハイアベイラビリティの基本単位はHAペアです。各ペアには、NVRAMデータのレプリケーションをサポートするための冗長リンクが含まれています。NVRAMは書き込みキャッシュではありません。コントローラ内部のRAMは書き込みキャッシュとして機能します。NVRAMの目的は、予期しないシステム障害から保護するためにデータを一時的にジャーナルすることです。この点では、データベースのREDOログに似ています。

NVRAMとデータベースのRedoログはどちらもデータを迅速に格納するために使用されるため、データに対する変更をできるだけ迅速にコミットできます。ドライブ（またはデータファイル）上の永続的データの更新は、ONTAPソフトウェアとほとんどのデータベースプラットフォームの両方で、チェックポイントと呼ばれるプロセスが実行されるまで行われません。通常動作中は、NVRAMデータおよびデータベースREDOログが読み取られません。

コントローラで突然障害が発生した場合、ドライブにまだ書き込まれていない保留中の変更がNVRAMに保存されている可能性があります。パートナーコントローラが障害を検出してドライブを制御し、NVRAMに保存されている必要な変更を適用します。

テイクオーバーとギブバック

テイクオーバーとギブバックは、HAペアのノード間でストレージリソースの責任を移すプロセスです。テイクオーバーとギブバックには次の2つの側面があります。

- ドライブへのアクセスを許可するネットワーク接続の管理
- ドライブ自体の管理

CIFSおよびNFSトラフィックをサポートするネットワークインターフェイスには、ホームロケーションとフェイルオーバーロケーションの両方が設定されます。テイクオーバーでは、ネットワークインターフェイスを元の場所と同じサブネット上にある物理インターフェイス上の一時ホームに移動します。ギブバックでは、ネットワークインターフェイスを元の場所に戻します。必要に応じて、正確な動作を調整できます。

iSCSIやFCなどのSANブロックプロトコルをサポートしているネットワークインターフェイスは、テイクオーバーやギブバックの実行時に再配置されません。代わりに、完全なHAペアを含むパスを使用してLUNをプロビジョニングする必要があります。これにより、プライマリパスとセカンダリパスが作成されます。

注： 大規模なクラスタ内のノード間でのデータの再配置をサポートするように追加のコントローラへのパスを設定することもできますが、この手順はHAプロセスの一部ではありません。

テイクオーバーとギブバックの2つ目の側面は、ディスク所有権の移行です。具体的なプロセスは、テイクオーバー/ギブバックの理由や実行したコマンドラインオプションなど、複数の要因によって異なります。目標は、できるだけ効率的に操作を実行することです。全体的なプロセスには数分かかるように見えるかもしれませんが、ドライブの所有権がノードからノードに移行される実際の瞬間は、通常数秒で測定できます。

テイクオーバー時間

テイクオーバー処理やギブバック処理の実行中にホストI/Oが短時間中断されますが、正しく設定された環境ではアプリケーションが停止することはありません。I/Oが遅延する実際の移行プロセスは数秒で測定されますが、ホストがデータパスの変更を認識してI/O処理を再送信するまでに、さらに時間がかかる場合があります。

中断の内容はプロトコルによって異なります。

- NFSおよびCIFSトラフィックをサポートするネットワークインターフェイスは、新しい物理的な場所への移行後に、ネットワークに対してAddress Resolution Protocol（ARP；アドレス解決プロトコル）要

求を発行します。この手順により、ネットワークスイッチはMACアドレステーブルを更新し、I/Oの処理を再開します。計画的なテイクオーバーとギブバックが実行される際のシステム停止は秒単位で測定され、検出されないことがよくあります。ネットワークによっては、ネットワークパスの変更を完全に認識するのに時間がかかる場合があります。また、OSによっては、かなり短時間に多数のI/Oがキューに登録され、再試行が必要になる場合があります。この余剰により、I/Oの再開に必要な時間が長くなる可能性があります。

- SANプロトコルをサポートするネットワークインターフェイスが新しい場所に移行されない。ホストOSが使用中のパスを変更する必要があります。ホストで検出されるI/Oの一時停止は、複数の要因によって異なります。ストレージシステムの観点から見ると、I/Oを処理できない時間はわずか数秒です。ただし、ホストOSによっては、I/Oがタイムアウトしてから再試行するまでに時間がかかることがあります。新しいOSではパスの変更をより迅速に認識できますが、古いOSでは通常、変更を認識するのに最大30秒かかります。

表1に、ストレージシステムがデータベース環境にデータを提供できない場合の想定テイクオーバー時間を示します。

表1）想定されるテイクオーバー時間

	NAS	SAN最適化OS	SAN
計画的なテイクオーバー	15秒	2～10秒	2~10秒
計画外のテイクオーバー	30 秒	2～15秒	30 秒

ONTAPソフトウェア

ONTAPは高度なデータ保護と管理の基盤をなすものですが、ソフトウェアのみで構成されます。ONTAPは、選択したプラットフォームで実行できます。

- AFFおよびFAS上のONTAP
- Cloud Volumes ONTAP
- ハイパースケーラクラウドプロバイダ向けCloud Volumes Services
- Azure NetApp Files

ここで重要となるのは、どのプラットフォームを選択したとしてもONTAPはONTAPであるという点です。あるプラットフォームはパフォーマンスに優れ、別のプラットフォームは低コストを実現します。ハイパースケーラ クラウド内で実行されるプラットフォームもあります。ONTAPのコア機能は変更されておらず、複数のレプリケーションオプションを使用して複数のONTAPシステムを1つの解決策にバインドできます。そのため、パフォーマンス要件、CAPEX（設備投資）/ OPEX（運用コスト）に関する考慮事項、全体的なクラウド戦略など、実際のニーズに基づいてデータ保護とディザスタリカバリの戦略を策定できます。基盤となるストレージ テクノロジは環境や場所を問わずどこでも動作します。

ONTAPとAll Flash FASおよびFASコントローラ

ONTAP搭載のAFF / FAS物理コントローラは、パフォーマンスとデータの制御性という点では最も優れたソリューションです。このオプションは、20年以上にわたって何千人もの顧客が信頼してきた標準です。ONTAPは、ミッションクリティカルな3つのデータベース、6万のデータベースサービスプロバイダ環境、ペタバイト規模のデータベースの瞬時のリストア処理、単一のデータベースの数百のクローンを使用するデータベースサービス（DBaaS）など、あらゆる環境に対応するソリューションを提供します。

Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAPはONTAPですが、ハイパースケーラクラウド環境で動作し、ハイパースケーラストレージボリュームにインテリジェンスとデータファブリックへの接続を提供します。ONTAPでOracleを実行する際のベストプラクティスには影響はありません。主な考慮事項は、パフォーマンスと、多少ですがコストです。

Cloud Volumes ONTAPのパフォーマンスは、クラウド プロバイダが管理する基盤のボリュームのパフォーマンスから部分的な制約を受けますが、その結果、ストレージの管理が容易になり、ONTAP Cloudのキャッシュ機能によってパフォーマンスが向上する場合もあります。ただしIOPSとレイテンシに関してはパブリック クラウド プロバイダ次第なので、多少の制約が常に発生します。これらの制限は、データベースのパフォーマンスが許容できないことを意味するものではありません。これは、単に、実際の物理AFFシステムなどの他のオプションよりもパフォーマンスの上限が低いことを意味します。さらに、Cloud Volumes ONTAPで使用されているさまざまなクラウドプロバイダが提供するストレージボリュームのパフォーマンスも継続的に向上しています。

現在、Cloud Volumes ONTAPの主なユースケースは開発とテスト作業ですが、一部のお客様は本番環境でもCloud Volumes ONTAPを使用しています。注目すべきレポートの1つに、Oracle Database In-Memory機能を使用してストレージのパフォーマンスに関する制限を緩和したことがあります。この機能を使用すると、データベースサーバをホストしている仮想マシンのRAMに、より多くのデータを格納できるため、ストレージのパフォーマンス要件が軽減されます。

クラウドでのネイティブNetApp

パブリック クラウド ストレージは多数ありますが、パフォーマンス、制御性、拡張性に限りがあるものがほとんどです。データベースワークロードに関する主な制限事項は次のとおりです。

- 多くのパブリック クラウド ストレージ オプションでは、最新のデータベース ワークロードに必要なIOPSレベルまで拡張したくても、コストや効率性、管理性のために拡張できない。
- パブリック クラウド プロバイダのIOPS機能が物理的に要件を満たしていても、多くの場合、I/Oレイテンシがデータベース ワークロードの要件に合わない。データベースがオールフラッシュストレージアレイに移行され、企業がレイテンシをミリ秒単位ではなくマイクロ秒単位で測定するようになったことで、この非互換性はさらに深刻になっています。
- パブリック クラウド ストレージの可用性は概ね優れているが、ミッションクリティカルな環境の厳しい要件を満たせるほどではない。
- パブリック クラウド ストレージ サービスにもバックアップとリカバリ機能があるが、大半のデータベースに求められるゼロのRPOやほぼゼロのRTOを達成できることはほとんどない。データ保護には、クラウド内のどこかとの間でバックアップとリカバリをストリーミングするのではなく、Snapshotベースの真の瞬時のバックアップとリカバリが必要です。

こうした制約を克服するために、Amazon Web Services（AWS）、Google Cloud Platform（GCP）、Azureなどの大手クラウドプロバイダは、それぞれのクラウド内に実際のNetAppハードウェアを基盤としたサーバを構築しています。お客様は、IOPS密度が高く、レイテンシの影響を受けやすいワークロード向けに設計されたストレージを使用して、ミッションクリティカルなデータベースインフラをクラウドに構築できます。さらに、お客様はネイティブクラウド環境で高度なONTAPデータ保護機能を活用できます。

また、一部のNetAppのお客様は、独自のイニシアチブでこのモデルを使用しています。例えばよく見るのが、自社のデータセンター施設から、ハイパースケール クラウド プロバイダの1つにいつでも高速アクセスできるようにする使い方です。別の例では、ハイパースケール クラウド プロバイダへの高速アクセスに、その機能を備えたコロケーション施設を使用しているお客様もいらっしゃいます。この手法により、Amazon AWS、Azure、IBM Cloudを、使用量に応じたオンデマンドの仮想サーバソースとして使用するようになりました。場合によっては、お客様の日常業務に何の変化もないこともあります。単純に、従来の仮想インフラに代わる、強力かつ柔軟でコスト効率に優れた方法としてハイパースケーラ サービスを利用する場合です。

ONTAPデータ保護の基礎

NetApp Snapshotコピーによるデータ保護

ONTAPデータ保護ソフトウェアの基盤は、NetApp Snapshot™ テクノロジです。主な値は次のとおりです。

- **シンプル**：Snapshotコピーは、特定の時点のデータコンテナの内容の読み取り専用コピーです。

- **効率性**：Snapshotコピーの作成時にスペースは必要ありません。スペースが消費されるのは、データが変更されたときだけです。
- **管理性**：SnapshotコピーはストレージOSに標準搭載されているため、Snapshotコピーに基づくバックアップ戦略を簡単に設定および管理できます。ストレージシステムの電源がオンになっていれば、バックアップを作成できます。
- **拡張性**ファイルとLUNの単一コンテナの最大1024個のバックアップを保持できます。複雑なデータセットの場合、データの複数のコンテナを、整合性のある単一のSnapshotコピーセットで保護できます。
- ボリュームに250個のSnapshotコピーが含まれているかどうかに関係なく、パフォーマンスには影響しません。

多くのストレージベンダーがSnapshotテクノロジを提供していますが、ONTAPのSnapshotテクノロジは革新的で、エンタープライズアプリケーションやデータベース環境に次のような大きなメリットをもたらします。

- ONTAP Snapshotコピーは、基盤となるWrite-Anywhere File Layout（WAFL®）の一部です。アドオンや外部テクノロジではありません。このレイアウトにより、ストレージシステムがバックアップシステムであるため、管理が簡単になります。
- ONTAP Snapshotコピーはパフォーマンスに影響しません。ただし、Snapshotコピーに大量のデータが格納されて基盤となるストレージシステムがいっぱいになる場合など、一部のエッジケースを除きます。
- 「整合グループ」という用語は、整合性のあるデータの集合として管理されるストレージオブジェクトをグループ化したものを指す場合によく使用されます。特定のONTAPボリュームのSnapshotコピーが整合グループのバックアップになります。

ONTAP Snapshotコピーの拡張性は、競合他社のテクノロジよりも優れています。パフォーマンスに影響を与えることなく、5、50、500個のスナップショットを保存できます。ボリュームに現在許可されているSnapshotコピーの最大数は1024です。Snapshotコピーの追加の保持が必要な場合は、Snapshotコピーを追加のボリュームにカスケードするオプションがあります。

そのため、ONTAPで実行されているデータベースの保護はシンプルで拡張性に優れています。データベースバックアップでは、データの移動は必要ありません。したがって、バックアップ戦略は、ネットワーク転送速度、多数のテープドライブ、ディスクステージング領域の制限ではなく、ビジネスのニーズに合わせて調整できます。

ONTAP SnapRestoreによるデータのリストア

NetApp SnapRestore® テクノロジを使用して、ONTAPでSnapshotコピーからデータを迅速にリストアできます。主な値は次のとおりです。

- 個々のファイルやLUNは、2TBのLUNでも4KBのファイルでも、数秒でリストアできます。
- LUNやファイルのコンテナ（NetApp FlexVol® ボリューム）全体を、10GBまたは100TBのデータであれ、数秒でリストアできます。

重要なデータベースが停止すると、重要なビジネスの運用が停止します。テープが破損する可能性があり、ディスクベースのバックアップからリストア処理を実行する場合でも、ネットワーク経由での転送に時間がかかることがあります。SnapRestoreでは、データベースをほぼ瞬時にリストアできるため、このような問題を回避できます。ペタバイト規模のデータベースでも、わずか数分で完全にリストアできます。

SnapRestoreがこれほど迅速かつ効率的に機能するのは、Snapshotコピーの性質によるものです。Snapshotコピーは、本質的には、特定の時点におけるボリュームの内容を並行して読み取り専用で表示する機能です。アクティブブロックは変更可能な実際のブロックですが、Snapshotコピーは、Snapshotコピーが作成された時点でのファイルおよびLUNを構成するブロックの状態を読み取り専用で表示します。

ONTAPでは、スナップショットデータへの読み取り専用アクセスのみが許可されますが、SnapRestoreを使用してデータを再アクティブ化できます。Snapshotコピーがデータの読み取り/書き込みビューとして再度有効になり、データは以前の状態に戻ります。SnapRestoreは、ボリュームレベルまたはファイルレベルで動作できます。この技術は基本的に同じで、動作に若干の違いがあります。

ボリューム対応のSnapRestore

ボリュームベースのSnapRestoreは、データのボリューム全体を以前の状態に戻します。この処理ではデータの移動は必要ありません。つまり、リストアプロセスはほぼ瞬時に完了しますが、APIまたはCLI処理の処理には数秒かかることがあります。1GBのデータをリストアするのは、1PBのデータをリストアするのと同じくらい複雑で時間のかかる作業ではありません。この機能は、多くのデータベースユーザがONTAPストレージシステムに移行する主な理由です。大規模なデータセットでも数秒でRTOを達成できます。

ボリュームベースSnapRestoreテクノロジの欠点の1つは、ボリューム内の変更が時間の経過とともに累積されることが原因です。したがって、各Snapshotコピーとアクティブなファイルデータは、その時点までの変更に依存します。ボリュームを以前の状態にリバートすると、データに対する以降の変更がすべて破棄されます。ただし、あまり明白ではありませんが、この反転にはあとで作成されたSnapshotコピーも含まれているため、必ずしも望ましくない場合があります。

たとえば、データ保持のSLAで夜間バックアップを30日間指定するとします。ボリュームベースのSnapRestoreを使用して5日前に作成されたSnapshotコピーにデータベースをリストアすると、過去5日間に作成されたSnapshotコピーがすべて破棄され、SLAに違反します。

この制限に対処するには、いくつかのオプションを使用できます。

1. ボリューム全体のSnapRestoreを実行するのではなく、以前のSnapshotコピーからデータをコピーできます。この方法は、小規模なデータセットで最適に機能します。
2. Snapshotコピーは、リストアの代わりにクローニングできます。この方法では、ソースSnapshotコピーがクローンの依存関係になるという制限があります。したがって、クローンも削除されるか、独立したボリュームにスプリットされないかぎり、削除することはできません。
3. ファイルベースのSnapRestoreを使用できます。

File SnapRestore

ファイルベースのSnapRestoreは、Snapshotベースのより詳細なリストアプロセスです。ボリューム全体の状態をリバートする代わりに、個々のファイルまたはLUNの状態がリバートされます。Snapshotコピーを削除する必要はありません。また、この処理によって以前のSnapshotコピーとの依存関係が作成されることもありません。ファイルまたはLUNがアクティブボリュームですぐに使用可能になります。

ファイルまたはLUNのSnapRestoreリストア中は、データを移動する必要はありません。ただし、ファイルまたはLUNの基盤となるブロックがSnapshotコピーとアクティブボリュームの両方に存在するようになったことを反映するには、一部の内部メタデータの更新が必要になります。パフォーマンスへの影響はありませんが、このバックグラウンド処理が完了するまでSnapshotコピーは作成できません。処理速度は約5GBps（18TB/時）です。これは、リストアするファイルの合計サイズに基づきます。

データレプリケーションとディザスタリカバリ

ほぼすべてのデータベースでデータレプリケーションが必要です。最も基本的なレベルでは、レプリケーションとは、オフサイトに保管されたテープ上のコピー、またはスタンバイデータベースへのデータベースレベルのレプリケーションを意味します。ディザスタリカバリとは、サービスが壊滅的に失われた場合に、これらのレプリカコピーを使用してサービスをオンラインにすることです。

ONTAPは、さまざまな要件にストレージアレイ内でネイティブに対応するための複数のレプリケーションオプションを提供し、幅広いニーズに対応します。これらのオプションには、同じプラットフォームでディザスタリカバリと高可用性の両方を実現する、完全に自動化された同期解決策まで、リモートサイトへのシンプルなバックアップレプリケーションが含まれます。

データベースに適用される主なONTAPレプリケーション機能は、NetApp SnapMirror® とNetApp SyncMirror® です。これらの機能はアドオン製品ではなく、ONTAPに完全に統合され、ライセンスキーを追加するだけでアクティブ化されます。また、ストレージレベルのレプリケーションだけが選択肢ではありません。Oracle DataGuardなどのデータベースレベルのレプリケーションを、ONTAPに基づくデータ保護戦略に統合することもできます。

適切な選択は、特定のレプリケーション、リカバリ、保持の要件によって異なります。

ONTAP SnapMirror

SnapMirrorは非同期レプリケーション解決策です。データベースやその関連アプリケーションなど、複雑で動的な大規模データセットの保護に最適です。主な値は次のとおりです。

- **管理性**：SnapMirrorは、ストレージソフトウェアに標準で組み込まれているため、設定と管理が容易です。アドオン製品は必要ありません。レプリケーション関係は数分で確立でき、ストレージシステム上で直接管理できます。
- **シンプル**：レプリケーションは、単一の整合グループとしてレプリケートされるLUNまたはファイルのコンテナであるFlexVolボリュームに基づいています。
- **効率性**：最初のレプリケーション関係が確立されると、変更内容のみがレプリケートされます。さらに、重複排除や圧縮などの効率化機能が維持されるため、リモートサイトに転送するデータ量がさらに削減されます。
- **柔軟性**：ミラーを一時的に解除してディザスタリカバリ手順をテストすれば、完全な再ミラーリングを必要とせずにミラーリングを簡単に再確立できます。ミラーを同期状態に戻すには、変更されたデータのみを適用する必要があります。ミラーリングを反転することで、災害が終了して元のサイトが稼働状態に戻ったあとに迅速に再同期することもできます。最後に、レプリケートされたデータの読み取り/書き込みクローンをテストと開発に使用できます。

ONTAPにはさまざまなレプリケーション機能がありますが、最も柔軟性に優れたのは、ボリューム間の非同期ミラーリングオプションであるSnapMirrorです。

前述したように、FlexVolボリュームは、Snapshotベースのバックアップと SnapRestoreベースのリカバリの基本的な管理単位です。FlexVolボリュームは、SnapMirrorベースのレプリケーションの基本単位でもあります。最初に、ソースボリュームからデスティネーションボリュームへのベースラインミラーを確立します。このミラー関係を初期化すると、以降のすべての処理は変更されたデータのみのレプリケーションに基づいて行われます。

データベースリカバリの観点から見ると、SnapMirrorの主な価値は次のとおりです。

- SnapMirrorの操作は理解しやすく、簡単に自動化できます。
- SnapMirrorレプリカを単純に更新するには、差分の変更のみをレプリケートする必要があるため、帯域幅の需要が減り、更新頻度が向上します。
- SnapMirrorは非常にきめ細かな機能です。シンプルなボリューム間の関係に基づいているため、独立して管理される数百個のレプリカやレプリケーション間隔を作成できます。レプリケーションを万能である必要はありません。
- 変更だけに基づいて関係を更新する機能を維持しながら、ミラーリングの方向を簡単に反転できます。このオプションを使用すると、災害（停電など）後にプライマリサイトが復旧したあとに、迅速なフェイルバック機能が提供されます。変更のみをソースに同期化する必要があります。
- ミラーは簡単に破損し、効率的に再同期することで、ディザスタリカバリ手順のリハーサルを行うことができます。
- SnapMirrorはフルブロックレベルのレプリケーションモードで動作し、ボリューム内のデータだけでなくSnapshotコピーもレプリケートします。この機能は、ディザスタリカバリサイトにデータのコピーと完全なバックアップセットの両方を提供します。
- SnapMirrorはバージョンに依存しないモードで動作するため、特定のSnapshotコピーをレプリケートでき、プライマリサイトとセカンダリサイトで異なる保持期間を設定できます。

SnapMirror Synchronous

NetApp SnapMirror Synchronous（SM-S）は、RPO=0の同期レプリケーションを提供するSnapMirrorの機能拡張です。同期ミラーリングが必要なのは、データ全体のサブセットのみのストレージアーキテクチャでよく使用されます。

SM-Sは、SyncとStrictSyncの2つのわずかに異なるモードで動作できます。

同期モードでは、確認される前に変更がレプリケートされます。この手順は、レプリケーションが稼働しているかぎり、RPOをゼロにすることを保証します。変更を複製できない場合、SM-Sは同期モードを終了し、操作を続行できます。この機能により、通常の状況ではRPO=0になりますが、レプリケーションデスティネーションが使用できない場合でもデータプロセスが完全に停止するわけではありません。

StrictSyncはRPO = 0を保証します。変更のレプリケートに失敗するとI/Oエラーが発生し、アプリケーションがシャットダウンします。

SM-Sの詳細については、TR-4733およびONTAPの公式ドキュメントを参照してください。ONTAPの新しいバージョンでは、機能が継続的に追加されています。

MetroClusterとSyncMirror

MetroClusterは、ミッションクリティカルな大規模ワークロードを対象とした同期レプリケーション機能でもあります。レプリケーションはSyncMirrorをベースにしており、最もシンプルなレイヤであるSyncMirrorは、RAIDで保護されたデータの完全なセットを2つの異なる場所に作成します。データセンター内の隣接する部屋に配置することも、数キロメートル離れた場所に配置することもできます。

SyncMirrorはONTAPと完全に統合されており、RAIDレベルのすぐ上で動作します。そのため、Snapshotコピー、SnapRestore、NetApp FlexClone® など、ONTAPの通常の機能はすべてシームレスに動作します。これはONTAPであり、同期データミラーリングの追加レイヤが含まれているだけです。

SyncMirrorデータを管理するONTAPコントローラの集まりをNetApp MetroCluster構成と呼びます。MetroClusterの主な目的は、一般的な障害やディザスタリカバリのさまざまな障害シナリオにおいて、同期ミラーリングされたデータへの高可用性アクセスを提供することです。

MetroClusterとSyncMirrorを使用したデータ保護の主な価値は次のとおりです。

- 通常運用時には、SyncMirrorは複数のサイト間の同期ミラーリングを保証します。書き込み処理は、両方のサイトの不揮発性メディアに存在するまで確認応答されません。
- サイト間の接続に障害が発生すると、SyncMirrorは自動的に非同期モードに切り替わり、接続が回復するまでプライマリサイトがデータを提供し続けます。リストア時には、プライマリサイトに蓄積された変更を効率的に更新することで、迅速な再同期を実現します。完全な再初期化は必要ありません。

SnapMirrorは、SyncMirrorベースのシステムとも完全に互換性があります。たとえば、プライマリデータベースが2つの地理的なサイトに分散したMetroClusterクラスタで実行されているとします。このデータベースは、長期アーカイブやDevOps環境でのクローン作成のために、バックアップを第3のサイトにレプリケートすることもできます。

MetroClusterでのOracleの使用に関する完全な概要については、TR-4592を参照してください。

整合グループ

「コンシステンシグループ」とは、ストレージアレイが複数のストレージリソースを単一のイメージとして管理できることを指します。たとえば、データベースが10個のLUNで構成されているとします。アレイは、これらの10個のLUNを一貫した方法でバックアップ、リストア、およびレプリケートできる必要があります。バックアップ時点でLUNのイメージに一貫性がない場合、リストアは実行できません。これらの10個のLUNをレプリケートするには、すべてのレプリカが相互に完全に同期されている必要があります。

ONTAPについて説明するときに整合性グループという用語を使用することはあまりありません。これは、ONTAPソフトウェアのボリュームとアグリゲートのアーキテクチャでは、整合性が常に基本的な機能であるためです。他の多くのストレージアレイは、LUNまたはファイルシステムを個別のユニットとして管理します。LUNは、データ保護を目的とした整合グループとしてオプションで構成することもできます。

ONTAPは、常に一貫性のあるローカルイメージとレプリケートされたデータイメージをキャプチャすることができました。ONTAPシステム上のさまざまなボリュームは、正式には整合グループとは呼ばれませんが、それが整合グループです。このボリュームのSnapshotコピーは整合グループのイメージであり、そのSnapshotコピーのリストアは整合グループのリストアです。SnapMirrorとNetApp SnapVault® はどちらも 整

合グループのレプリケーションを提供します。

従属書き込み順序

技術的な観点から見ると、整合性グループの鍵となるのは、書き込み順序（特に従属書き込み順序）を維持することです。たとえば、10個のLUNに書き込むデータベースは、すべてのLUNに同時に書き込みます。多くの書き込みは非同期で発行されます。つまり、書き込みが完了する順序は重要ではありません。実際に実行される順序は、オペレーティングシステムとネットワークの動作によって異なります。

データベースが追加の書き込みを続行するには、一部の書き込み処理がディスク上に存在している必要があります。このような重要な書き込み処理は、依存書き込みと呼ばれます。以降の書き込みI/Oは、これらの書き込みがディスクに存在するかどうかに左右されます。これら10個のLUNのスナップショット、リカバリ、またはレプリケーションでは、従属書き込み順序が保証されていることを確認する必要があります。ファイルシステムの更新は、書き込み順序に依存する書き込みのもう1つの例です。ファイルシステムの変更の順序を維持する必要があります。そうしないと、ファイルシステム全体が破損する可能性があります。

注： 一部のストレージシステムでは、「非同期」処理を設定できます。つまり、永続的メディアにコミットされる前に書き込みの確認応答を受け取ることができます。この機能によりパフォーマンスは向上しますが、ストレージシステムのクラッシュや停電が発生した場合のデータ損失は事実上保証されます。ONTAPはこのように設定することはできません。常に同期モードで動作します。ONTAPによって確認応答された書き込みは、永続的メディアに格納されています。

整合性グループのSnapshot

整合グループSnapshot（CG Snapshot）は、ONTAPの基本的なSnapshotテクノロジを拡張したものです。標準のSnapshot処理では、1つのボリューム内のすべてのデータの整合性のあるイメージが作成されますが、複数のボリューム間、さらには複数のストレージシステム間で整合性のある一連のSnapshotを作成する必要がある場合があります。その結果、1つのボリュームのSnapshotと同じ方法で使用できる一連のSnapshotが作成されます。ローカルデータのリカバリに使用することも、ディザスタリカバリの目的でレプリケートすることも、単一の一貫したユニットとしてクローニングすることもできます。

CG Snapshotの最大の用途は、12台のコントローラにまたがる約1PBのデータベース環境です。このシステムで作成されたCG Snapshotは、バックアップ、リカバリ、クローニングに使用されています。

ほとんどの場合、データセットが複数のボリュームにまたがっていて書き込み順序を保持する必要がある場合、選択した管理ソフトウェアによってCG Snapshotが自動的に使用されます。このような場合、CGスナップショットの技術的な詳細を理解する必要はありません。ただし、複雑なデータ保護要件がある場合は、データ保護とレプリケーションのプロセスを詳細に管理する必要があります。自動化ワークフローや、カスタムスクリプトを使用してCG Snapshot APIを呼び出すこともできます。最適なオプションとCG Snapshotの役割を理解するには、テクノロジの詳細な説明が必要です。

CG Snapshotセットの作成は、次の2つのプロセスで行います。

1. すべてのターゲットボリュームで書き込みフェンシングを確立します。
2. フェンシングされた状態のボリュームのSnapshotを作成します。

書き込みフェンシングはシリアルで確立されます。つまり、フェンシングプロセスは複数のボリュームにまたがってセットアップされるため、書き込みI/Oは最初のボリュームでフリーズされ、以降に表示されるボリュームにコミットされたままになります。このプロセスは、最初は書き込み順序を維持するための要件に違反しているように見えるかもしれませんが、環境ホストで非同期的に実行され、他の書き込みには依存しません。

たとえば、データベースはデータファイルの非同期更新を多数問題し、OSがI/Oの順序を変更して、独自のスケジューラ設定に従って完了できるようにします。アプリケーションとオペレーティングシステムが書き込み順序を保持する要件をすでにリリースしているため、このタイプのI/Oの順序は保証できません。

カウンタの例として、ほとんどのデータベースロギングアクティビティは同期です。I/Oが確認応答され、書き込み順序を維持する必要があるまで、データベースはログへの以降の書き込みを続行しません。ログI/Oがフェンシングされたボリュームに到達した場合、そのことは確認されず、アプリケーションはそれ以降の書き

込みをブロックします。同様に、ファイルシステムのメタデータI/Oは通常同期です。たとえば、ファイル削除処理が失われることはありません。`xfs` ファイルシステムを使用するオペレーティングシステムがファイルを削除し、xfsファイルシステムのメタデータを更新してそのファイルへの参照を削除したI/Oがフェンシングされたボリュームにある場合、ファイルシステムのアクティビティは一時停止します。この一時停止により、CG Snapshot処理中のファイルシステムの整合性が保証されます。

ターゲットボリューム間で書き込みフェンシングを設定すると、それらのボリュームでSnapshotを作成できるようになります。ボリュームの状態は従属書き込みの観点から凍結されるため、Snapshotを正確に同時に作成する必要はありません。cgスナップショットを作成するアプリケーションの欠陥を防ぐために、初期書き込みフェンシングには設定可能なタイムアウトが含まれています。このタイムアウトでは、ONTAPが自動的にフェンシングを解除し、定義された秒数後に書き込み処理を再開します。タイムアウト時間の経過前にすべてのSnapshotが作成された場合、作成される一連のSnapshotは有効な整合グループになります。

NetApp SnapCenterソフトウェアおよびその他の管理ツール

データベース環境におけるONTAPの主な価値は、瞬時のSnapshotコピー、シンプルなSnapMirrorレプリケーション、効率的なFlexCloneボリュームの作成など、ONTAPのコアテクノロジにあります。ONTAPでこれらの機能を簡単に設定して要件を満たすこともありますが、より複雑なニーズに対応するにはオーケストレーションレイヤが必要です。

このTRの目的は、データベースストレージアーキテクチャをSnapshotと親和性が高く、すぐに使用できるようにすることを目的として、ONTAPでのデータ保護の原則を説明することです。

SnapCenter

NetApp SnapCenter® は、NetAppの主力データ保護製品です。データベースバックアップの実行方法という点では、SnapManager® 製品と似ています。違いは、NetAppストレージシステム上のデータ保護管理を単一コンソールで管理できるように一から設計されている点です。

SnapCenterには、Snapshotベースのバックアップやリストア処理、SnapMirror、SnapVaultレプリケーションなどの基本機能や、大企業の大規模な運用に必要なその他の機能が含まれています。これらの高度な機能には、拡張されたロールベースアクセス制御（RBAC）機能、サードパーティのオーケストレーション製品と統合するためのRESTful API、データベースホストでSnapCenterプラグインを無停止で一元管理、クラウド規模環境向けに設計されたUIなどがあります。

データ保護計画

ローカルデータベースのデータ保護アーキテクチャ

適切なデータベースデータ保護アーキテクチャは、データの保持、リカバリ性、さまざまなイベント発生時のシステム停止に対する耐性に関するビジネス要件によって異なります。

たとえば、スコープ内のデータベースの数を考えてみましょう。管理するオブジェクトが少ないため、単一データベースのバックアップ戦略を構築し、一般的なSLAへの準拠を確保することは簡単です。データファイルのセットとログファイルのセットは1つだけです。データベースの数が増えるにつれて、監視が複雑になり、データベース管理者（DBA）はバックアップの失敗に対処するために多くの時間を費やすことになる可能性があります。データベース環境がクラウドに到達し、サービスプロバイダが拡張するにつれて、まったく異なるアプローチが必要になります。

データベースのサイズも戦略に影響します。100GBのデータベースのバックアップとリカバリには、データセットが非常に小さいため、多くのオプションがあります。従来のツールを使用してバックアップメディアからデータをコピーするだけで、リカバリに十分なRTOが得られます。一方、100TBのデータベースでは、RTOで数日間の停止が許容される場合を除き、通常はまったく異なる戦略が必要です。その場合は、従来のコピーベースのバックアップおよびリカバリの手順で十分かもしれません。

最後に、バックアップとリカバリのプロセス自体以外にも考慮が必要な要素があります。たとえば、データベースが重要な本番環境のアクティビティをサポートしているため、熟練したDBAだけがリカバリを実行するまれなイベントになっているとしますか。あるいは、データベースは、リカバリが頻繁に発生し、ジェネラリストのITチームが管理する大規模な開発環境に含まれていますか。

これらの考慮事項は、データ保護戦略の選択に影響します。次のセクションでは、リレーショナルデータベースプラットフォームに適用される基本原則について説明します。

Snapshotはバックアップですか？

データ保護戦略としてSnapshotを使用することに対する一般的な反対意見の1つは、「実際の」データとSnapshotデータが同じドライブに配置されていることです。これらのドライブが失われると、プライマリデータとバックアップの両方が失われます。

この問題は有効な問題です。ローカルSnapshotは、日々のバックアップとリカバリのニーズに使用され、その点でSnapshotはバックアップです。NetApp環境のすべてのリカバリシナリオの99%近くが、最も厳しいRTO要件を満たすためにSnapshotを使用しています。

ただし、ローカルSnapshotを唯一のバックアップ戦略にすることはできません。このため、NetAppには、SnapMirrorやSnapVaultレプリケーションなどのテクノロジが搭載されており、独立したドライブセットにSnapshotを迅速かつ効率的にレプリケートできます。スナップショットとスナップショットレプリケーションを使用して適切に設計された解決策では、テープの使用を最小限に抑えて四半期ごとのアーカイブを作成することも、完全に排除することもできます。

ハツクアツフトリカハリノカイヨウ

次の表2 に、ローカルデータベース保護の基本オプションとその利点と制限事項をまとめます。このテーブルは、同期ミラーリングのデータ保護には対応していません。この要件については、TR-4592：『Oracle on MetroCluster』を含むNetApp MetroClusterのドキュメントを参照してください。

表2）ローカルデータ保護の概要

	整合グループ	ログのバックアップと再生
ローカルリカバリのRPO	1時間 （15分可能）	秒
ローカルリカバリのRTO	秒	分
拡張性	高いRPOが許容される大量のデータベースに最適なオプション	非常に大規模なデータベースに対応する最大限の柔軟性

次のセクションでは、これらのオプションについて詳しく説明します。

整合グループのバックアップ

整合グループのバックアップでは、データベース（複数のデータベース）と関連するアプリケーションの状態を1つのアトミックポイントインタイムでキャプチャします。この1つのキャプチャには、データファイル、ログファイル、データベースに直接関連付けられているその他のファイルなど、すべてのデータベースコンポーネントが含まれます。このプロセスは、Oracle RDBMS、Microsoft SQL Server、SAP HANA、PostgreSQL、 MySQL、MariaDBなどがあります。

データベース環境全体のSnapshotを作成する場合、基本的にはクラッシュをシミュレートするため、このようなバックアップはcrash-consistentバックアップと呼ばれることがよくあります。リカバリシナリオのサポートに関して懸念が生じることがありますが、リカバリ手順は不要であることを理解しておくことが重要です。

整合グループバックアップのリストア後にデータベースを起動すると、通常のログリカバリプロセスが実行され、バックアップ時点で転送中だったI/Oが再生されます。データベースが起動します。

基本的に、データの不整合なしに電源障害やサーバクラッシュに耐えることができるすべてのデータベースをこの方法で保護できます。これは、さまざまなベンダーが提供する同期ミラーリング製品や非同期ミラーリング製品で保護されている膨大な数のデータベースでも実証されています。プライマリサイトで突然災害が発生した場合、レプリカサイトには、災害発生時の元のデータベースの整合性のあるイメージが格納されます。繰り返しますが、特別なリカバリ手順は必要ありません。データベースのサバイバーコピーを使用してデータベースを起動すると、ログが自動的に再生され、データベースが開きます。

通常、このアプローチのRPOはバックアップの時点までに制限されます。通常、シングルボリュームデータベースSnapshotの最小RPOは1時間です。たとえば、時間単位のSnapshotを48個、夜間のSnapshotを30日間追加することは妥当であり、大量のSnapshotを保存する必要はありません。RPOを1時間未満に抑えることは難しくなります。 環境、拡張性、データ保護の要件を理解するために、まずNetAppプロフェッショナルサービスに相談しないと、この目標を達成することはできません。

通常、RTOは数秒で測定できます。データベースがシャットダウンされ、ボリュームがリストアされ、データベースが再起動されます。少量のログ再生が発生し、データベースがオンラインになります。

保持期間は選択したRPOに関連付けられています。長期間の保持できめ細かなRPOを維持すると、スナップショットが大量に作成され、ストレージプラットフォームの制限を超える可能性があります。詳細については、前述のRPOの制限事項を参照してください。

最も簡単な方法は、すべてのファイルまたはLUNを1つのボリューム整合性グループに配置することです。これにより、ONTAPで直接Snapshotの作成をスケジュールできます。データベースが複数のボリュームにまたがる必要がある場合は、整合性グループのSnapshotコピー（CG Snapshot）が必要です。SnapCenterと旧世代のSnap Centerでは、定義済みのボリュームリスト上にシンプルな整合性グループSnapshotを作成できます。また、スケジューリング、運用前/運用後のスクリプト作成機能、およびレプリケーション管理も含まれます。

データベースが複数のボリュームにまたがる必要がある場合は、整合性グループのSnapshotコピー（CG Snapshot）が必要です。整合性のあるSnapshotの作成に使用される最も一般的なソフトウェアは、NetApp Snap Creator® です。このソフトウェアは、有効なサポート契約があるすべてのコントローラで無償で利用できます。Snap Creatorには、スケジュール設定、処理の前後のスクリプト作成機能、レプリケーション管理機能もあります。SnapCenter Plug-in for Oracle Databaseなどの製品では、基盤となるデータセットで必要な場合にCGスナップショットもネイティブで実行されます。最後に、CGスナップショットは、さまざまなスクリプト言語でNetApp Manageability SDKを使用して簡単にスクリプト化できます。

ログのバックアップと再生

ログベースのバックアップは、ポイントインタイムリカバリ機能と可能な限り低いRPOを必要とする重要なデータベースに最適なオプションです。このプロセスは、従来のテープまたはファイルベースのバックアッププロセスに基づいているため、ほとんどのDBAにとって馴染みのあるプロセスです。違いは、データファイルのコピープロセスがスナップショットによって置き換えられる点です。ほぼ瞬時に処理されるだけでなく、データベースサーバ、ストレージシステム、ネットワーク上でのデータ移動による負荷も解消されます。

以降のセクションでは、特定のデータベースプラットフォームの正確なバックアッププロセスについて説明しますが、通常は同じ手順に従います。

1. これで、データベースはバックアップ手順に対応できるようになります。手順は異なります。
2. データファイルのSnapshotが作成されます。データファイルが複数のボリュームにまたがっている場合は、整合性グループSnapshotが必要になることがあります。
3. ログファイルのSnapshotが作成されます。

その結果、以下を含む一連のSnapshotコピーが作成されます。

- データファイルのリカバリ可能なイメージを含むSnapshotコピー。

- データベースの整合性を確保するために必要なログファイルのSnapshotコピー。

このアプローチのRPOは、通常の状況ではゼロです。データベースのリカバリ状況のほとんどは、ユーザまたはアプリケーションのエラーによってデータベースが破損したり、データベースが実際に破損した可能性が低いことが原因です。リカバリでは、データファイルのみをリストアし、ディスクに残っているログファイルを使用してデータベースを目的の時点に戻す必要があります。これがRPOゼロの現在の状態です。

ログファイルも破損した場合、ログファイルのスナップショットの頻度を増やすと、データ損失を最小限に抑えることができます。不正な管理者がファイルを積極的に削除しようとすると、データ損失の可能性を完全に排除することは不可能ですが、被害を最小限に抑えることができます。

たとえば'rm -rf /がデータ・ファイルとログ・ファイルの両方を削除した場合は'両方のスナップショットをリカバリする必要がありますログファイルのSnapshot作成頻度が1時間に設定されている場合、この災害に近い状況におけるRPOは1時間になります。このレベルのデータ保護を実現するには、大量のデータ移動を必要としないSnapshotのようなテクノロジが欠かせません。

RTOは、データファイルのSnapshotの頻度によって効果的に制御されます。たとえば、データファイルのSnapshotが24時間ごとに作成された場合、最悪のRTOシナリオでは、前回のSnapshotから23時間59分後に障害が発生します。データベースを完全にリカバリするには、約24時間分のログファイルをバックアップに適用する必要があります。生成されるログの量や使用するリレーショナルデータベース管理システムによっては、このプロセスの完了に5分から24時間かかることがあります。データログの再生に必要な時間が許容できない場合は、データファイルのスナップショット作成頻度を増やすことができます。

保持期間には、フルデータベースバックアップとログファイルバックアップという2つの個別に制御されるバックアップがあるため、2つの側面があります。一般に、データベースでは一定期間のポイントインタイムリカバリ機能が必要ですが、ポイントオブザバックアップリカバリの方が広くなります。一般的な例として、データベースを毎晩バックアップし、それらの夜間Snapshotを90日間保持することがあります。また、ログファイルは7日間保持される場合があります。その結果、保持期間が90日のデータベースが作成されますが、特定のポイントインタイムリカバリは直前の7日以内にしか実行できません。

レプリケーションとディザスタリカバリのアーキテクチャ

表3 は、セキュアなオフサイトストレージとディザスタリカバリを目的として、データをリモートサイトにレプリケートするリモートデータ保護について説明しています。このテーブルは、同期ミラーリングのデータ保護には対応していません。この要件については、TR-4592『Oracle on MetroCluster』を含むNetApp MetroClusterのドキュメントを参照してください。

表3）レプリケーションとディザスタリカバリ

	整合グループ	ログレプリケーション	データベースレプリケーション
ディザスタリカバリRPO	1時間（15分可能）	ゼロ（SnapMirror Synchronous）から数分（非同期SnapMirror）	ゼロから分
ディザスタリカバリのRTO	秒	分	秒
拡張性	高いRPOが許容される大量のデータベースに最適なオプション	非常に大規模なデータベースに対応する最大限の柔軟性	RPOは低いが拡張性は低い少数のデータベースに適している

整合グループのレプリケーションは、整合グループのバックアップをレプリケートするプロセスです。整合性グループには、データファイル、ログファイル、データベースに直接関連付けられているその他のファイルなど、すべてのデータベースコンポーネントを含める必要があります。アプリケーションデータを含めることもできます。

RPOは、使用可能なネットワーク帯域幅と保護対象のデータベースの合計サイズによって制限されます。初回のベースライン転送が作成されたあとは、変更されたデータのみに基づいて更新されます。変更されたデー

タは、一般にデータベースの合計サイズに占める割合が低くなります。一般的な原則として、1時間に1回データベースを更新することは可能です。使用可能な帯域幅に基づいて制限があります。

たとえば、週単位の変更率が10%の10TBデータベースでは、1時間あたりの合計変更量は約6GBです。10Gbの接続では、このデータベースの転送に約6分かかります。変更率はデータベースの変更率の変動に応じて変化しますが、全体的には更新間隔が15分であるため、RPOが15分になるはずです。このようなデータベースが100個ある場合、データの転送に600分かかります。したがって、1時間のRPOは不可能です。同様に、100TBの単一データベースのレプリカのサイズが週に10%の変更率であれば、1時間で確実に更新することはできません。

レプリケーションのオーバーヘッドや同時レプリケーション処理数の制限など、レプリケーションに影響するその他の要因があります。ただし、単一ボリュームのレプリケーション戦略の全体的なプランニングは、使用可能な帯域幅に基づいて行うことができ、レプリケーションRPOを1時間にすることができます。RPOが1時間未満になると達成が難しくなるため、NetAppプロフェッショナルサービスに相談してから実行する必要があります。サイト間のネットワーク接続が非常に良好な場合は、15分で完了することがあります。ただし、全体的に1時間未満のRPOが必要な場合は、マルチボリュームのログ再生アーキテクチャが適しています。

ディザスタリカバリシナリオにおける整合グループレプリケーションを使用したRTOは非常に優れており、ストレージの観点から見れば、通常は数秒で測定されます。最も簡単な方法は、単にミラーを解除することであり、データベースを起動する準備ができています。通常、データベースの起動時間は約10秒ですが、大量のトランザクションがログに記録されている非常に大規模なデータベースには数分かかることがあります。

RTOを決定するうえで最も重要な要素は、ストレージシステムではなく、ストレージシステムを実行するアプリケーションとホストオペレーティングシステムです。たとえば、レプリケートされたデータベースデータを1秒または2秒で使用できるようにすることができますが、このレプリカはデータのみを表します。また、データを使用できるアプリケーションバイナリを備えた、適切に構成されたオペレーティングシステムも必要です。

お客様は、オペレーティングシステムで事前に検出されたストレージを使用して、ディザスタリカバリインスタンスを事前に準備している場合もあります。このような場合、災害復旧シナリオをアクティブ化するには、ミラーを解除してデータベース・サーバを起動するだけです。また、OSと関連するアプリケーションをデータベースと一緒にESX Virtual Machine Disk（VMDK；仮想マシンディスク）としてミラーリングする場合もあります。このような場合のRPOは、データベースを起動できるようにVMDKをブートするためにお客様がどれだけ自動化に投資したかによって決まります。

保持期間の一部はSnapshotの制限によって制御されます。たとえば、ONTAPのボリュームのSnapshotコピー数には上限が255個あります。場合によっては、レプリケーションを多重化して制限を増やすこともあります。たとえば、500日分のバックアップが必要な場合は、ソースを2つのボリュームにレプリケートし、別の日に更新を実行できます。このセットアップでは、必要な初期スペースを増やす必要がありますが、フルバックアップを複数回実行する従来のバックアップシステムよりもはるかに効率的なアプローチです。

単一ボリュームの整合グループ

最も簡単な方法は、すべてのファイルまたはLUNを1つのボリューム整合グループに配置することです。これにより、SnapMirrorおよびSnapVaultの更新をストレージシステム上で直接スケジュールできます。外部ソフトウェアは必要ありません。

マルチボリューム整合グループ

データベースが複数のボリュームにまたがっている必要がある場合は、整合性グループSnapshot（CG Snapshot）が必要です。繰り返しになりますが、整合性のあるスナップショットをレプリケートするために使用される最も一般的なソフトウェアはOnCommand Snap Creator Frameworkです。Snap Creatorには、スケジュール設定、処理の前後のスクリプト作成機能、レプリケーション管理機能もあります。SnapCenterなどの製品は、基盤となるデータセットで必要に応じてCGスナップショットをネイティブで実行します。

また、ディザスタリカバリを目的としたマルチボリュームの整合性のあるSnapshotの使用についても、もう1つ考慮すべき点があります。複数のボリュームの更新を実行すると、転送の進行中に災害が発生する可能性があります。その結果、一連のボリュームが互いに整合性のない状態になります。ボリューム間に整合性がない場合

は、crash-consistentで使用可能なデータベースイメージを提供するために、一部のボリュームを以前のSnapshot状態にリストアする必要があります。

ログレプリケーション

ログレプリケーションアプローチは、ポイントインタイムリカバリ機能と可能な限り低いRPOを必要とする重要なデータベースに最適な方法です。また、低いRPOを維持するためにログファイルのみを短時間でレプリケートする必要があるため、帯域幅効率にも優れています。このプロセスは、データファイルとログファイルが分離されたバックアップ手順です。データファイルとログファイルは、さまざまなスケジュールでレプリケートされます。

基本的なプロセスは、ローカルバックアップの実行と同じです。

1. これで、データベースはバックアップ手順に対応できるようになります。手順は異なります。
2. データファイルのSnapshotが作成されます。データファイルが複数のボリュームにまたがっている場合は、整合性のあるグループSnapshotが必要になることがあります。
3. ログファイルのSnapshotが作成されます。

次のスナップショットタイプが作成されます。

- データファイルのリカバリ可能なイメージを含むSnapshot。
- データベースの整合性を確保するために必要なログファイルのスナップショット。

次に、レプリケーションスケジュールを個別に設定し、RPOとRTOを制御します。

- RPOは、ログファイルの更新頻度によって決まります。
- RTOは、データファイルの更新頻度によって制御されます。

たとえば、100TBのデータベースでRPOが15分、RTOが1時間の場合を考えてみましょう。一般的な構成では、データファイルレプリカが1日に1回更新され、ログファイルレプリカが15分ごとに更新されます。災害が発生すると、ミラーが解除され、使用可能なすべてのログが再生されます。最悪のシナリオは、前回のデータファイルの更新から23時間59分後に発生する災害です。ログは23時間45分再生され、レプリケートされていないログデータは15分失われます。

このアプローチのRPOは、使用可能な帯域幅によって制限されます。ほとんどの場合、非常に大規模なデータベースであっても1時間のRPOを達成でき、優れたネットワークインフラであれば15分を達成できます。15分未満の間隔でレプリケーションを実行することは可能ですが、データベースログの生成が通常変動するため、信頼性が低下する傾向があります。多くの時間を5分おきにレプリケートすることも可能ですが、更新の間に書き込まれたログデータの量がわずか5分では移動できない場合があります。RPO = 0が必要な場合は、ログデータにSnapMirror Synchronousを使用できます。

RTOは、データファイルの更新頻度によって効果的に制御されます。たとえば、データファイルのSnapshotが24時間ごとに更新される場合、最悪のRTOシナリオは、前回のバックアップから23時間59分後に障害が発生したことになります。データベースを完全にリカバリするには、約24時間分のログファイルをバックアップに適用する必要があります。このリカバリには、生成されるログの量や使用するリレーショナルデータベース管理システムによっては、5分から24時間かかることがあります。データログの再生に必要な時間が許容できない場合は、データファイルを24時間から12時間に短縮できます。

保持期間には、フルデータベースバックアップとログファイルバックアップという2つの個別に制御されるバックアップがあるため、2つの側面があります。一般に、データベースでは一定期間のポイントインタイムリカバリ機能が必要ですが、ポイントオブザバックアップリカバリの方が広くなります。一般的な例として、データベースを夜間にバックアップし、夜間のバックアップを90日間保持することがあります。また、ログファイルは7日間保持される場合があります。その結果、保持期間が90日のデータベースが作成されますが、特定のポイントインタイムリカバリは、過去7日間の期間内にしか実行できません。

ディザスタリカバリ：アクティブ化

NFS

ディザスタリカバリコピーをアクティブ化するプロセスは、ストレージのタイプによって異なります。NFSでは、ファイルシステムをディザスタリカバリサーバに事前にマウントできます。これらは読み取り専用状態であり、ミラーが解除されると読み取り/書き込みになります。このタスクによってRPOが非常に低くなり、管理するパーツが少なくなるため、ディザスタリカバリプロセス全体の信頼性が向上します。

SAN

ディザスタリカバリの一環としてSAN構成をアクティブ化することは、より複雑になります。最も簡単な方法は、ミラーを一時的に解除してSANリソースをマウントする方法です。この手順には、論理ボリュームマネージャ（LVM）構成の検出（Oracle Automatic Storage Management[ASM]などのアプリケーション固有の機能を含む）、へのエントリの追加などの手順が含まれます `/etc/fstab`。

その結果、LUNのデバイスパス、ボリュームグループ名、およびその他のデバイスパスがターゲットサーバに認識されます。その後、これらのリソースをシャットダウンし、その後ミラーをリストアできます。その結果、サーバはデータベースストレージを迅速にオンラインにできる状態になります。ボリュームグループ、ファイルシステムのマウント、またはASMインスタンスのアクティブ化の手順は、データベース自体を起動するのと同じスクリプトで簡単に自動化できます。

ディザスタリカバリ環境が最新の状態であることを確認します。たとえば、新しいLUNがソースサーバに追加される可能性があります。つまり、ディザスタリカバリプランが想定どおりに機能するように、デスティネーションで新しいLUNを事前に検出しておく必要があります。

Oracle整合グループのバックアップ

Oracleデータベース全体（データ・ファイル、アーカイブ・ログ、REDOログ、制御ファイルなど）を1つのボリュームに配置 は、有効なバックアップ、リストア、およびレプリケーション方法です。ただし、RPOはバックアップ自体のポイントに制限されます。1時間以上のRPOに適しています。データベースが複数のボリュームにまたがっている場合は、前述のいずれかのツールを使用してCG Snapshotを作成する必要があります。

たとえば、次のSnapshotスケジュールを使用して、データベース全体を1つのボリュームに含めることができます。

- 72時間ごとのスナップショット
- 30個の夜間Snapshot
- 12の月次スナップショット

このスケジュールでは、過去72時間の間に1時間のRPOを達成し、夜間および月次バックアップを追加で作成します。単一のボリュームまたは一連のCG Snapshotに複数のデータベースやアプリケーションファイルを含めることもでき、大規模な環境で一貫したバックアップを実現できます。

Oracleのホットバックアップ

ホット バックアップからリカバリするには、次の2組のデータが必要です。

- バックアップモードでのデータファイルのスナップショット
- データファイルがホットバックアップモードのときに作成されたアーカイブログ

コミットされたすべてのトランザクションを含む完全なリカバリが必要な場合は、3つ目の項目が必要です。

- 最新のREDOログ

ホットバックアップのリカバリを促進する方法はいくつかあります。多くのお客様は、ONTAP CLIを使用してSnapshotをリストアし、次にOracle RMANまたはsqlplusを使用してリカバリを完了します。この方法は、データベースをリストアする可能性と頻度が非常に低く、熟練したデータベース管理者が対応する大規模な本

番環境では特に一般的です。完全な自動化を実現するために、NetApp SnapCenterなどのソリューションには、コマンドラインインターフェイスとグラフィカルインターフェイスの両方を備えたOracleプラグインが含まれています。

大規模なお客様の中には、スケジュールされたSnapshotに備えて特定の時間にデータベースをホットバックアップモードにするように、ホストで基本的なスクリプトを設定することで、よりシンプルなアプローチを採用しているお客様もいます。たとえば、alter database begin backup at 23：58、alter database end backup at 00：02というコマンドのスケジュールを設定してから、ストレージシステム上で直接Snapshotを作成するようにスケジュールを設定します。その結果、外部のソフトウェアやライセンスを必要としない、シンプルで拡張性に優れたバックアップ戦略が実現します。

データレイアウト

最もシンプルなレイアウトは、データファイルを1つ以上の専用ボリュームに分離することです。これらのファイルは、他のファイルタイプによって汚染されていない必要があります。これは、重要なREDOログ、制御ファイル、またはアーカイブログを削除することなく、SnapRestore処理によってデータファイルボリュームを迅速にリストアできるようにするためです。

SANには、専用ボリューム内でのデータファイルの分離に関する同様の要件があります。Microsoft Windowsなどのオペレーティングシステムでは、1つのボリュームに複数のデータファイルLUNが含まれ、それぞれにNTFSファイルシステムが設定される場合があります。他のオペレーティング・システムでは'通常'論理ボリューム・マネージャが使用されますたとえば、Oracle ASMでは、ASMディスクグループのLUNを1つのボリュームに限定し、1つのボリュームとしてバックアップおよびリストアできるようにするのが最も簡単なオプションです。パフォーマンスまたは容量管理のために追加のボリュームが必要な場合は、新しいボリュームに追加のディスクグループを作成すると、管理が簡単になります。

これらのガイドラインに従うと、整合性グループSnapshotを実行する必要なく、ストレージシステム上で直接Snapshotをスケジュールできます。これは、Oracleのホットバックアップでは、データファイルを同時にバックアップする必要がないためです。ホットバックアップ手順は、データファイルが数時間にわたってテープにゆっくりとストリーミングされても、継続的に更新されるように設計されています。

ASMディスクグループを複数のボリュームに分散して使用すると、複雑な状況が発生します。このような場合は、CG Snapshotを実行して、すべてのコンスティチュエントボリュームでASMメタデータの整合性を確保する必要があります。

注意： ASM `spfile` と `passwd` ファイルが、データファイルをホストするディスクグループに含まれていないことを確認してください。これらのファイルがデータファイルをホストしているディスクグループにある場合、データファイルとデータファイルのみを選択的にリストアする機能が妨げられます。

ローカルリカバリ手順–NFS

この手順は、手動で実行することも、SnapCenterなどのアプリケーションを使用して実行することもできます。基本的な手順は次のとおりです。

1. データベースをシャットダウンします。
2. 目的のリストアポイントの直前に、データファイルボリュームをSnapshotにリカバリします。
3. アーカイブログを目的のポイントまで再生します。
4. 完全なリカバリが必要な場合は、現在のREDOログを再生します。

この手順では、目的のアーカイブログがアクティブファイルシステムにまだ存在していることを前提としています。サポートされていない場合は、アーカイブログをリストアするか、または `rman/sqlplus` Snapshotディレクトリ内のデータに転送する必要があります。

また、小規模なデータベースの場合は `.snapshot`、自動化ツールやストレージ管理者が `snaprestore` コマンドを実行しなくても、エンドユーザがディレクトリから直接データファイルをリカバリできます。

ローカルリカバリ手順–SAN

この手順は、手動で実行することも、SnapCenterなどのアプリケーションを使用して実行することもできます。基本的な手順は次のとおりです。

1. データベースをシャットダウンします。
2. データファイルをホストしているディスクグループを休止します。手順は、選択した論理ボリュームマネージャによって異なります。ASMでは、このプロセスでディスクグループをディスマウントする必要があります。Linuxでは、ファイルシステムをディスマウントし、論理ボリュームと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。目的は、リストア対象のターゲットボリュームグループに対するすべての更新を停止することです。
3. 目的の復元ポイントの直前に、データファイルのディスクグループをスナップショットに復元します。
4. 新しくリストアしたディスクグループを再アクティブ化します。
5. アーカイブログを目的のポイントまで再生します。
6. 完全なリカバリが必要な場合は、すべてのREDOログを再生します。

この手順では、目的のアーカイブログがアクティブファイルシステムにまだ存在していることを前提としています。サポートされていない場合は、アーカイブログLUNをオフラインにしてリストアを実行し、アーカイブログをリストアする必要があります。この手順も、アーカイブログを専用ボリュームに分割すると便利です。アーカイブログがRedoログとボリュームグループを共有している場合は、LUNのセット全体をリストアする前にRedoログを他の場所にコピーする必要があります。この手順により、最終的に記録されたトランザクションの損失を防ぐことができます。

Oracle Snapshot向けに最適化されたバックアップ

Oracle 12cでは、データベースをホットバックアップモードにする必要がないため、Snapshotベースのバックアップとリカバリがさらに簡単になります。Snapshotベースのバックアップをストレージ システム上で直接スケジュール設定することができますが、引き続き完全なリカバリまたはポイントインタイム リカバリを実行できます。

データベース管理者にとってはホットバックアップリカバリの手順の方がなじみがありますが、データベースがホットバックアップモードのときに作成されなかったSnapshotを使用することは以前からありました。ただしOracle 10gおよび11gでは、データベースの一貫性を保つためにリカバリ時に手動での操作が別途必要でした。Oracle 12cでは、 `sqlplus rman` ホットバックアップモードではないデータファイルバックアップに対してアーカイブログを再生するロジックがとに追加されています。

前述したように、スナップショットベースのホットバックアップをリカバリするには、次の2セットのデータが必要です。

- バックアップモードで作成されたデータファイルのSnapshot
- データファイルがホットバックアップモードのときに生成されたアーカイブログ

リカバリ中、データベースはデータファイルからメタデータを読み取り、リカバリに必要なアーカイブログを選択します。

Snapshotを使用して最適化されたリカバリでは、同じ結果を得るために必要なデータセットはわずかに異なります。

- データファイルのスナップショット、およびスナップショットが作成された時刻を識別する方法
- 最新のデータファイルチェックポイントの時刻からSnapshotの正確な時刻までのログをアーカイブします。

リカバリ中、データベースはデータファイルからメタデータを読み取り、必要な最も古いアーカイブログを特定します。フルリカバリまたはポイントインタイムリカバリを実行できます。ポイントインタイムリカバリを実行する場合は、データファイルのスナップショットの時刻を把握することが重要です。指定したリカバリポイントは、Snapshotの作成時刻以降である必要があります。NetAppでは、クロックの変動を考慮して、スナップショット時間に少なくとも数分を追加することを推奨しています。

詳細については、Oracle 12cの各種ドキュメントで、「Recovery Using Storage Snapshot Optimization」のトピックを参照してください。また、Oracleサードパーティ製スナップショットのサポートについては、OracleのドキュメントID Doc ID 604683.1を参照してください。

データレイアウト

最も簡単なレイアウトは、データファイルを1つ以上の専用ボリュームに分離する方法です。これらのファイルは、他のファイルタイプによって汚染されていない必要があります。この手順では、重要なREDOログ、制御ファイル、またはアーカイブログを削除することなく、SnapRestore処理でデータファイルボリュームを迅速にリストアできるようにします。

SANには、専用ボリューム内でのデータファイルの分離に関する同様の要件があります。Microsoft Windowsなどのオペレーティングシステムでは、1つのボリュームに複数のデータファイルLUNが含まれ、それぞれにNTFSファイルシステムが設定される場合があります。他のオペレーティング・システムでは'通常'論理ボリューム・マネージャも使用されますたとえば、Oracle ASMでは、ディスクグループを1つのボリュームに限定し、1つのボリュームとしてバックアップおよびリストアできるようにするのが最も簡単なオプションです。パフォーマンスまたは容量管理のために追加のボリュームが必要な場合は、新しいボリュームに追加のディスクグループを作成すると、管理が容易になります。

これらのガイドラインに従うと、整合性グループSnapshotを実行することなく、ONTAPで直接Snapshotをスケジュールできます。これは、Snapshotで最適化されたバックアップでは、データファイルを同時にバックアップする必要がないためです。

ASMディスクグループが複数のボリュームに分散されている場合は、複雑な問題が発生します。このような場合は、CG Snapshotを実行して、すべてのコンスティチュエントボリュームでASMメタデータの整合性を確保する必要があります。

注意： ASM `spfile` と `passwd` ファイルが、データファイルをホストするディスクグループに含まれていないことを確認してください。この配置は、データファイルおよびデータファイルのみを選択的にリストアする機能を妨げます。

ローカルリカバリ手順–NFS

この手順は、手動で実行することも、SnapCenterなどのアプリケーションを使用して実行することもできます。基本的な手順は次のとおりです。

1. データベースをシャットダウンします。
2. 目的のリストアポイントの直前に、データファイルボリュームをSnapshotにリカバリします。
3. アーカイブログを目的のポイントまで再生します。

この手順では、目的のアーカイブログがアクティブファイルシステムにまだ存在していることを前提としています。サポートされていない場合は、アーカイブログをリストアするか、 `rman sqlplus .snapshot` ディレクトリ内のデータに転送する必要があります。

また、小規模なデータベースの場合は `.snapshot`、エンドユーザがディレクトリから直接データファイルをリカバリできます。自動化ツールやストレージ管理者によるSnapRestoreコマンドの実行は不要です。

ローカルリカバリ手順–SAN

この手順は、手動で実行することも、SnapCenterなどのアプリケーションを使用して実行することもできます。基本的な手順は次のとおりです。

1. データベースをシャットダウンします。
2. データファイルをホストしているディスクグループを休止します。手順は、選択した論理ボリュームマネージャによって異なります。ASMでは、このプロセスでディスクグループをディスマウントする必要があります。Linuxでは、ファイルシステムをディスマウントし、論理ボリュームと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。目的は、リストア対象のターゲットボリュームグループに対するすべての更新を停止することです。
3. 目的の復元ポイントの直前に、データファイルのディスクグループをスナップショットに復元します。

4. 新しくリストアしたディスクグループを再アクティブ化します。
5. アーカイブログを目的のポイントまで再生します。

この手順では、目的のアーカイブログがアクティブファイルシステムにまだ存在していることを前提としています。サポートされていない場合は、アーカイブログLUNをオフラインにしてリストア処理を実行し、アーカイブログをリストアする必要があります。この操作は、アーカイブログを専用ボリュームに分割すると便利です。アーカイブログがRedoログとボリュームグループを共有している場合は、記録された最終的なトランザクションが失われないように、LUNセット全体のリストア前にRedoログを別の場所にコピーする必要があります。

フルリカバリの例

データファイルが破損または破壊されており、完全なリカバリが必要であると仮定します。そのための手順は次のとおりです。

```
[oracle@jfs2 ~]$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup mount;

ORACLE instance started.

Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.

SQL> recover automatic;
Media recovery complete.

SQL> alter database open;

Database altered.

SQL>
```

ポイントインタイムリカバリの例

リカバリ手順全体は単一のコマンドです `recover automatic`。

ポイントインタイムリカバリが必要な場合は、Snapshotのタイムスタンプがわかっている必要があり、次のように特定できます。

```
EcoSystems-8060::> snapshot show -vserver svm0 -volume NTAP_oradata -fields create-time

vserver   volume        snapshot   create-time
--------- ----------------- ------------ --------------------------------
svm0      NTAP_oradata my-backup Thu Mar 09 10:10:06 2017
```

Snapshotの作成時間は3月9日と10：10：06と表示されます。安全のために、Snapshotの時刻に1分が追加されます。

```
[oracle@jfs2 ~]$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup mount;

ORACLE instance started.

Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
```

```
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.

SQL> recover database until time '09-MAR-2017 10:44:15' snapshot time '09-MAR-2017 10:11:00';
```

リカバリが開始されました。スナップショット時間は、記録された時間の1分後の10：11：00、目標復旧時間は10：44と指定されています。次に、sqlplusは目的のリカバリ時間（10：44）に到達するために必要なアーカイブログを要求します。

```
ORA-00279: change 551760 generated at 03/09/2017 05:06:07 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_31_930813377.dbf
ORA-00280: change 551760 for thread 1 is in sequence #31


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 552566 generated at 03/09/2017 05:08:09 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_32_930813377.dbf
ORA-00280: change 552566 for thread 1 is in sequence #32


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 553045 generated at 03/09/2017 05:10:12 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_33_930813377.dbf
ORA-00280: change 553045 for thread 1 is in sequence #33


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

ORA-00279: change 753229 generated at 03/09/2017 05:15:58 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_34_930813377.dbf
ORA-00280: change 753229 for thread 1 is in sequence #34


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}

Log applied.
Media recovery complete.

SQL> alter database open resetlogs;

Database altered.

SQL>
```

注： `recover automatic` コマンドを使用したSnapshotを使用したデータベースの完全なリカバリに `snapshot time` は特定のライセンスは必要ありませんが、を使用したポイントインタイムリカバリにはOracle Advanced Compressionライセンスが必要です。

整合グループのディザスタリカバリ

整合グループのレプリケーションは、単一のVolume SnapMirrorのレプリケーションをスケジュール設定するだけで簡単に実行できます。この単一ボリュームレプリカには、データファイル、制御ファイル、アーカイブログ、REDOログが含まれます。SnapMirrorの更新を行うたびに、デスティネーションサイトにデータベースの新しいコピーが作成され、ミラーを解除することで整合性があり、アクティブ化できる状態になります。

データベースが複数のボリュームにまたがる必要がある場合は、整合性グループSnapshot（CG Snapshot）が必要です。追加情報のCG Snapshotコピーの管理については、「ログのレプリケーション」セクションを参照してください。

ブロックレベルレプリケーションモードのSnapMirrorでこの戦略を使用すると、ソースストレージシステム上のすべてのSnapshotの完全なレプリケーションが可能になります。ディザスタリカバリコピーに加えて、バックアップ全体がレプリケートされます。

ログ再生を使用したディザスタリカバリ

Oracleデータベースのレプリケーション手順は、基本的にバックアップ手順と同じです。主な要件は、リカバリ可能なバックアップを構成するSnapshotをリモートストレージシステムにレプリケートする必要があることです。

前述したローカルデータ保護のセクションで説明したように、リカバリ可能なバックアップは、ホットバックアッププロセスを使用して作成することも、Snapshotで最適化されたバックアップを利用して作成することもできます。

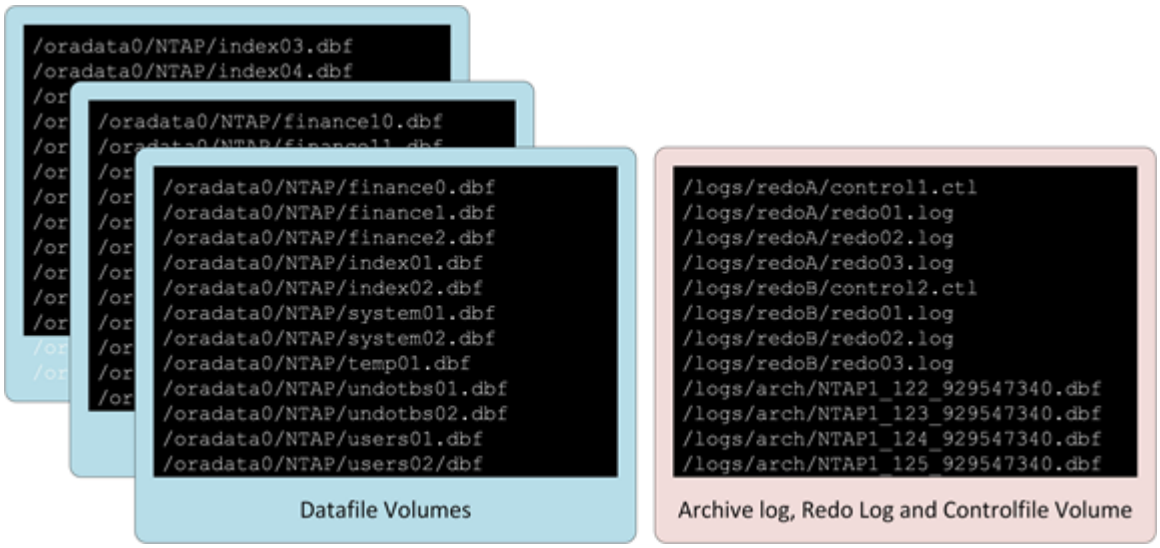
最も重要な要件は、データファイルを1つ以上の専用ボリュームに分離することです。これらのファイルは、他のファイルタイプによって汚染されていない必要があります。これは、データファイルのレプリケーションが、アーカイブログなどの他のデータタイプのレプリケーションから完全に独立していることを確認するためです。ファイルレイアウトの詳細、およびストレージレイアウトがスナップショットに適していることを確認するための重要な詳細については、「データレイアウト」セクションを参照してください。

データファイルが専用ボリュームにカプセル化されているとしたら、次の質問はREDOログ、アーカイブログ、制御ファイルをどのように管理するかです。最も簡単なアプローチは、これらすべてのデータタイプを1つのボリュームに配置することです。この方法の利点は、レプリケートされたREDOログ、アーカイブログ、制御ファイルが完全に同期されることです。不完全リカバリやバックアップ制御ファイルの使用は必須ではありませんが、その他のリカバリシナリオに備えてバックアップ制御ファイルの作成スクリプトを作成する方が望ましい場合もあります。

2ボリュームのレイアウト

最も単純なレイアウトを図2に示します。

図2）2ボリュームのレイアウト



このレイアウトは最も一般的なアプローチです。DBAの観点からは、Redoログとアーカイブログのすべてのコピーを同じボリュームに配置することは通常とは異なる場合があります。ただし、ファイルとLUNがすべて基盤となる同じドライブセットに配置されている場合、分離によって保護が強化されるわけではありません。

3ボリュームのレイアウト

データ保護に関する懸念や、RedoログのI/Oをコントローラ間で分散する必要があるために、Redoログを分離する必要がある場合があります。その場合は、図3に示す3つのボリュームからなるレイアウトを レプリケーションに使用しても、不完全リカバリを実行したり、バックアップ制御ファイルに依存したりする必要はありません。

図3）3ボリュームのレイアウト

このレイアウトにより、ソース上の独立したスピンドルとコントローラのセット全体でREDOログと制御ファイルをストライピングできます。ただし、アーカイブログ、制御ファイルおよびREDOログのセットは、アーカイブログと同期された状態でレプリケートできます。

このモデルでは、RedoログBボリュームはレプリケートされません。

ディザスタリカバリの手順–ホットバックアップ

ホットバックアップを使用してディザスタリカバリを実行するには、次の基本的な手順を使用します。

前提条件

1. Oracleバイナリはディザスタリカバリサーバにインストールされます。
2. データベースインスタンスは、 `/etc/oratab`.
3. `passwd pfile spfile` インスタンスのおよびまたはは、 `$ORACLE_HOME/dbs` ディレクトリに存在する必要があります。

ディザスタ リカバリ

1. データファイルと共通ログボリュームのミラーを解除します。
2. データファイルのボリュームを、データファイルの最新のホットバックアップSnapshotコピーにリストアします。
3. SANを使用する場合は、ボリュームグループをアクティブ化するか、ファイルシステムをマウントします。
4. アーカイブログを目的のポイントまで再生します。
5. 完全なリカバリが必要な場合は、現在のREDOログを再生します。

NFSを使用すると、データファイルとログファイル用のNFSファイルシステムをディザスタリカバリサーバにいつでもマウントできるため、手順が大幅に簡易化されます。ミラーが切断されると読み取り/書き込みになります。

ディザスタリカバリの手順–Snapshotに最適化されたバックアップ

スナップショット用に最適化されたバックアップからのリカバリは、ホットバックアップリカバリ手順とほぼ同じですが、次の点が異なります。

1. データファイルと共通ログボリュームのミラーを解除します。
2. 現在のログボリュームレプリカの前に作成されたSnapshotコピーにデータファイルボリュームをリストアします。
3. SANを使用する場合は、ボリュームグループをアクティブ化するか、ファイルシステムをマウントします。
4. アーカイブログを目的のポイントまで再生します。
5. 完全なリカバリが必要な場合は、現在のREDOログを再生します。

これらの違いにより、データベースがホットバックアップモードのときにソースでSnapshotが正しく作成されたことを確認する必要がないため、全体的なリカバリ手順が簡易化されます。ディザスタリカバリ手順は、ディザスタリカバリサイトのSnapshotのタイムスタンプに基づいています。スナップショット作成時のデータベースの状態は重要ではありません。

ホットバックアップSnapshotコピーによるディザスタリカバリ

次に、ホットバックアップSnapshotのレプリケーションに基づくディザスタリカバリ戦略の例を示します。また、シンプルで拡張性に優れたローカルバックアップ戦略の一例としても機能します。

この例のデータベースは、基本的な2ボリュームアーキテクチャに配置されています。`/oradata` データファイルが格納され、`/oralogs` REDOログ、アーカイブログ、および制御ファイルを組み合わせて使用されます。

```
[root@jfs2 ~]# ls /ora*

/oradata:
dbf

/oralogs:
arch ctrl redo
```

2つのスケジュールが必要です。1つは夜間のデータファイルバックアップ用、もう1つはログファイルバックアップ用です。これらの時間は、それぞれ真夜中と15分と呼ばれます。

```
EcoSystems-8060::> job schedule cron show -name midnight|15minutes
Name                 Description
---------------------    ------------------------------------------------------------------------
15minutes            @:00,:15,:30,:45
midnight             @0:00
2 entries were displayed.
```

`NTAP-datafile-backups NTAP- log-backups`次に示すように、これらのスケジュールはSnapshotポリシーおよび内で使用されます。

```
EcoSystems-8060::> snapshot policy show -vserver jfsCloud0 -policy NTAP-* -fields
schedules,counts
vserver   policy                schedules                      counts
------------ --------------------------- ------------------------------------ --------
jfsCloud0 NTAP-datafile-backups midnight                       60
jfsCloud0 NTAP-log-backups      15minutes                      72
2 entries were displayed.
```

最後に、これらのSnapshotポリシーがボリュームに適用されます。

```
EcoSystems-8060::> volume show -vserver jfsCloud0 -volume jfs2_oracle* -fields snapshot-policy
vserver   volume                   snapshot-policy
------------ --------------------------- ---------------------------
jfsCloud0 jfs2_oracle_datafiles NTAP-datafile-backups
jfsCloud0 jfs2_oracle_logs       NTAP-log-backups
```

このアプリケーションは、ボリュームのバックアップスケジュールを定義します。データファイルのSnapshotは午前0時に作成され、60日間保持されます。ログボリュームには、15分間隔で作成された72個のSnapshotコピーが格納されます。これにより、最大で18時間が経過します。

次に、データファイルのSnapshotが作成されたときにデータベースがホットバックアップモードになっていることを確認します。この手順では、指定したSIDでバックアップモードを開始および停止するいくつかの基本的な引数を受け入れる小さなスクリプトを使用します。

```
58 * * * * /snapomatic/current/smatic.db.ctrl --sid NTAP --startbackup
02 * * * * /snapomatic/current/smatic.db.ctrl --sid NTAP --stopbackup
```

次の手順では、午前0時のSnapshotを囲む4分間の間に、データベースがホットバックアップモードになります。

ディザスタリカバリサイトへのレプリケーションは次のように設定されます。

```
EcoSystems-8060::> snapmirror show -destination-path jfsCloud1:jfsdr2* -fields source-
path,destination-path,schedule
source-path                          destination-path                    schedule
------------------------------------ ----------------------------------- ----------
jfsCloud0:jfs2_oracle_datafiles jfsCloud1:jfsdr2_oracle_datafiles   6hours
jfsCloud0:jfs2_oracle_logs       jfsCloud1:jfsdr2_oracle_logs       15minutes
2 entries were displayed.
```

ログボリュームのデスティネーションは15分ごとに更新され、RPOは約15分になります。正確な更新間隔は、更新中に転送する必要があるデータの合計量によって少し異なります。

データファイルボリュームのデスティネーションは6時間ごとに更新されますが、この更新はRPOやRTOには影響しません。ディザスタリカバリが必要な場合は、最初にデータファイルボリュームをホットバックアップSnapshotにリストアします。更新間隔を短くする目的は、このボリュームの転送速度をスムーズにすることです。更新が1日に1回スケジュールされている場合は、その日に蓄積されたすべての変更を一度に転送する必要があります。更新頻度が高くなると、変更は1日のうちに徐々にレプリケートされます。

災害が発生した場合は、最初に両方のボリュームのミラーを解除します。

```
EcoSystems-8060::> snapmirror break -destination-path jfsCloud1:jfsdr2_oracle_datafiles -force
Operation succeeded: snapmirror break for destination "jfsCloud1:jfsdr2_oracle_datafiles".

EcoSystems-8060::> snapmirror break -destination-path jfsCloud1:jfsdr2_oracle_logs -force
Operation succeeded: snapmirror break for destination "jfsCloud1:jfsdr2_oracle_logs".

EcoSystems-8060::>
```

これで'レプリカは読み取り/書き込み可能になります次に、ログボリュームのタイムスタンプを確認します。

```
EcoSystems-8060::> snapmirror show -destination-path jfsCloud1:jfsdr2_oracle_logs -field newest-
snapshot-timestamp
source-path                  destination-path              newest-snapshot-timestamp
---------------------------- ----------------------------- ---------------------------
jfsCloud0:jfs2_oracle_logs jfsCloud1:jfsdr2_oracle_logs 03/14 13:30:00
```

ログボリュームの最新のコピーは3月14日13：30：00です。

次に、ログボリュームの状態の直前に作成されたホットバックアップSnapshotを特定します。ログ再生プロセスでは、すべてのアーカイブログがホットバックアップモードで作成される必要があるため、このタスクが必要です。したがって、ログボリュームレプリカはホットバックアップイメージよりも古いものである必要があります。そうしないと、必要なログが含まれません。

```
EcoSystems-8060::> snapshot list -vserver jfsCloud1 -volume jfsdr2_oracle_datafiles -fields
create-time -snapshot midnight*

vserver   volume                    snapshot                  create-time
```

```
jfsCloud1 jfsdr2_oracle_datafiles   midnight.2017-01-14_0000   Sat Jan 14 00:00:00 2017
jfsCloud1 jfsdr2_oracle_datafiles   midnight.2017-01-15_0000   Sun Jan 15 00:00:00 2017

...

jfsCloud1 jfsdr2_oracle_datafiles   midnight.2017-03-12_0000   Sun Mar 12 00:00:00 2017
jfsCloud1 jfsdr2_oracle_datafiles   midnight.2017-03-13_0000   Mon Mar 13 00:00:00 2017
jfsCloud1 jfsdr2_oracle_datafiles   midnight.2017-03-14_0000   Tue Mar 14 00:00:00 2017
60 entries were displayed.

EcoSystems-8060::>
```

最後に作成されたSnapshotはです midnight.2017-03-14_0000。このSnapshotコピーは、データファイルの最新のホットバックアップイメージであり、次のようにリストアされます。

```
EcoSystems-8060::> snapshot restore -vserver jfsCloud1 -volume jfsdr2_oracle_datafiles -snapshot
midnight.2017-03-14_0000

EcoSystems-8060::>
```

この段階で、データベースをリカバリする準備が整いました。データベースがSAN環境の場合は、次の手順でボリュームグループのアクティブ化とファイルシステムのマウントを行います。このプロセスは簡単に自動化できます。この例ではNFSを使用しているため、ファイルシステムはすでにマウントされており、読み取り/書き込み可能になります。ミラーが破損した瞬間にマウントやアクティブ化を行う必要はありません。

これで、データベースを任意の時点にリカバリすることも、レプリケートされたREDOログのコピーに完全にリカバリすることもできます。この例は、アーカイブログ、制御ファイル、およびREDOログボリュームを組み合わせた値を示しています。バックアップ制御ファイルやリセットログファイルに依存する必要がないため、リカバリプロセスが大幅に簡易化されます。

```
[oracle@jfsdr2 ~]$ sqlplus / as sysdba

Connected to an idle instance.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1090522752 bytes
Database Buffers          503316480 bytes
Redo Buffers               13848576 bytes
Database mounted.

SQL> recover database until cancel;

ORA-00279: change 1291884 generated at 03/14/2017 12:58:01 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_34_938169986.dbf
ORA-00280: change 1291884 for thread 1 is in sequence #34


Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
auto
ORA-00279: change 1296077 generated at 03/14/2017 15:00:44 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_35_938169986.dbf
ORA-00280: change 1296077 for thread 1 is in sequence #35
ORA-00278: log file '/oralogs_nfs/arch/1_34_938169986.dbf' no longer needed for
this recovery

...

ORA-00279: change 1301407 generated at 03/14/2017 15:01:04 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_40_938169986.dbf
ORA-00280: change 1301407 for thread 1 is in sequence #40
ORA-00278: log file '/oralogs_nfs/arch/1_39_938169986.dbf' no longer needed for
this recovery
```

```
ORA-00279: change 1301418 generated at 03/14/2017 15:01:19 needed for thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_41_938169986.dbf
ORA-00280: change 1301418 for thread 1 is in sequence #41
ORA-00278: log file '/oralogs_nfs/arch/1_40_938169986.dbf' no longer needed for
this recovery


ORA-00308: cannot open archived log '/oralogs_nfs/arch/1_41_938169986.dbf'
ORA-17503: ksfdopn:4 Failed to open file /oralogs_nfs/arch/1_41_938169986.dbf
ORA-17500: ODM err:File does not exist


SQL> recover database;

Media recovery complete.

SQL> alter database open;

Database altered.

SQL>
```

Snapshotで最適化されたバックアップによるディザスタリカバリ

Snapshotで最適化されたバックアップを使用したディザスタリカバリ手順は、ホットバックアップディザスタリカバリ手順とほぼ同じです。ホットバックアップSnapshot手順と同様に、ディザスタリカバリ用にバックアップをレプリケートするローカルバックアップアーキテクチャの拡張機能でもあります。次の例は、詳細な設定とリカバリ手順を示しています。この例では、ホットバックアップとSnapshotで最適化されたバックアップの主な違いも示しています。

この例のデータベースは、基本的な2ボリュームアーキテクチャに配置されています。/oradata データファイルが格納され、/oralogs REDOログ、アーカイブログ、および制御ファイルを組み合わせて使用されます。

```
 [root@jfs3 ~]# ls /ora*
/oradata:
dbf

/oralogs:
arch ctrl redo
```

2つのスケジュールが必要です。1つは夜間のデータファイルバックアップ用、もう1つはログファイルバックアップ用です。これらの時間は、それぞれ真夜中と15分と呼ばれます。

```
EcoSystems-8060::> job schedule cron show -name midnight|15minutes
Name                Description
-------------------- --------------------------------------------------------------------------
15minutes           @:00,:15,:30,:45
midnight            @0:00
2 entries were displayed.
```

NTAP-datafile-backups NTAP- log-backups次に示すように、これらのスケジュールはSnapshotポリシーおよび内で使用されます。

```
EcoSystems-8060::> snapshot policy show -vserver jfsCloud0 -policy NTAP-* -fields
schedules,counts
vserver   policy                schedules                     counts
---------- -------------------------- ------------------------------------ --------
jfsCloud0 NTAP-datafile-backups midnight                      60
jfsCloud0 NTAP-log-backups      15minutes                     72
2 entries were displayed.
```

最後に、これらのSnapshotコピーポリシーがボリュームに適用されます。

```
EcoSystems-8060::> volume show -vserver jfsCloud0 -volume jfs3_oracle* -fields snapshot-policy
vserver   volume                 snapshot-policy
```

```
------------ ---------------------------- ----------------------------
jfsCloud0 jfs2_oracle_datafiles NTAP-datafile-backups
jfsCloud0 jfs2_oracle_logs       NTAP-log-backups
```

ポリシーは、ボリュームの最終的なバックアップスケジュールを制御します。Snapshotは午前0時に作成され、60日間保持されます。ログボリュームには、15分間隔で作成された72個のSnapshotコピーが格納されます。これにより、最大で18時間が経過します。

ディザスタリカバリサイトへのレプリケーションは次のように設定されます。

```
EcoSystems-8060::> snapmirror show -destination-path jfsCloud1:jfsdr3* -fields source-
path,destination-path,schedule
source-path                        destination-path                    schedule
---------------------------------------- ---------------------------------------- -----------
jfsCloud0:jfs3_oracle_datafiles jfsCloud1:jfsdr3_oracle_datafiles 6hours
jfsCloud0:jfs3_oracle_logs       jfsCloud1:jfsdr3_oracle_logs       15minutes
2 entries were displayed.
```

ログボリュームのデスティネーションは15分ごとに更新されます。この更新により、RPOは約15分になります。正確な更新間隔は、更新中に転送する必要があるデータの合計量によって多少異なります。

データファイルのボリュームのデスティネーションは6時間ごとに更新されます。この更新は、RPOやRTOには影響しません。ディザスタリカバリが必要な場合は、まずデータファイルボリュームをホットバックアップSnapshotにリストアする必要があります。
更新間隔を短くする目的は、このボリュームの転送速度をスムーズにすることです。更新が1日に1回スケジュールされている場合は、その日に蓄積されたすべての変更を一度に転送する必要があります。更新頻度が高くなると、変更は1日のうちに徐々にレプリケートされます。

災害が発生した場合は、最初にすべてのボリュームのミラーを解除します。

```
EcoSystems-8060::> snapmirror break -destination-path jfsCloud1:jfsdr3_oracle_datafiles -force
Operation succeeded: snapmirror break for destination "jfsCloud1:jfsdr3_oracle_datafiles".

EcoSystems-8060::> snapmirror break -destination-path jfsCloud1:jfsdr3_oracle_logs -force
Operation succeeded: snapmirror break for destination "jfsCloud1:jfsdr3_oracle_logs".

EcoSystems-8060::>
```

これで'レプリカは読み取り/書き込み可能になります次に、ログボリュームのタイムスタンプを確認します。

```
EcoSystems-8060::> snapmirror show -destination-path jfsCloud1:jfsdr3_oracle_logs -field newest-
snapshot-timestamp
source-path                 destination-path             newest-snapshot-timestamp
--------------------------------- ---------------------------------- ---------------------------------
jfsCloud0:jfs3_oracle_logs jfsCloud1:jfsdr3_oracle_logs 03/14 13:30:00
```

ログボリュームの最新のコピーは3月14日13：30です。次に、ログボリュームの状態の直前に作成されたデータファイルのSnapshotを特定します。ログ再生プロセスでは、Snapshotの直前から目的のリカバリポイントまでのすべてのアーカイブログが必要になるため、この処理が必要になります。

```
EcoSystems-8060::> snapshot list -vserver jfsCloud1 -volume jfsdr3_oracle_datafiles -fields
create-time -snapshot midnight*

vserver   volume                    snapshot                    create-time
--------- ------------------------- --------------------------- ------------------------
jfsCloud1 jfsdr3_oracle_datafiles   midnight.2017-01-14_0000    Sat Jan 14 00:00:00 2017
jfsCloud1 jfsdr3_oracle_datafiles   midnight.2017-01-15_0000    Sun Jan 15 00:00:00 2017

...

jfsCloud1 jfsdr3_oracle_datafiles   midnight.2017-03-12_0000    Sun Mar 12 00:00:00 2017
jfsCloud1 jfsdr3_oracle_datafiles   midnight.2017-03-13_0000    Mon Mar 13 00:00:00 2017
jfsCloud1 jfsdr3_oracle_datafiles   midnight.2017-03-14_0000    Tue Mar 14 00:00:00 2017
60 entries were displayed.

EcoSystems-8060::>
```

最後に作成されたSnapshotはです `midnight.2017-03-14_0000`。このSnapshotをリストアします。

```
EcoSystems-8060::> snapshot restore -vserver jfsCloud1 -volume jfsdr3_oracle_datafiles -snapshot
midnight.2017-03-14_0000

EcoSystems-8060::>
```

これで、データベースをリカバリする準備が整いました。データベースがSAN環境の場合は、ボリュームグループのアクティブ化とファイルシステムのマウントが簡単に自動化されます。ただし、この例ではNFSを使用しているため、ファイルシステムはすでにマウントされており、読み取り/書き込み可能になっています。ミラーが破損した瞬間にマウントやアクティブ化を行う必要はありません。

これで、データベースを任意の時点にリカバリすることも、レプリケートされたREDOログのコピーに完全にリカバリすることもできます。この例は、アーカイブログ、制御ファイル、およびREDOログボリュームを組み合わせた値を示しています。バックアップ制御ファイルやリセットログファイルに依存する必要がないため、リカバリプロセスが大幅に簡易化されます。

```
[oracle@jfsdr3 ~]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Wed Mar 15 12:26:51 2017

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup mount;
ORACLE instance started.

Total System Global Area 1610612736 bytes
Fixed Size
Variable Size            1073745536 bytes
Database Buffers          520093696 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover automatic;
Media recovery complete.
SQL> alter database open;

Database altered.

SQL>
```

詳細情報の入手方法

このドキュメントに記載されている情報の詳細については、以下のドキュメントやWebサイトを確認してください。

- ONTAPとONTAP System Managerのドキュメント リソース
  https://www.netapp.com/data-management/oncommand-system-documentation/
- ONTAP 9ドキュメント センター
  https://docs.netapp.com/ontap-9/index.jsp
- NetAppの製品ドキュメント
  https://www.netapp.com/support-and-training/documentation/

本ドキュメントに記載されている製品や機能のバージョンがお客様の環境でサポートされるかどうかについては、NetApp サポート サイトで Interoperability Matrix Tool (IMT)を参照してください。NetApp IMT には、NetApp がサポートする構成を構築するために使用できる製品コンポーネントやバージョンが定義されています。サポートの可否は、お客様の実際のインストール環境が公表されている仕様に従っているかどうかによって異なります。

機械翻訳に関する免責事項

原文は英語で作成されました。英語と日本語訳の間に不一致がある場合には、英語の内容が優先されます。公式な情報については、本資料の英語版を参照してください。翻訳によって生じた矛盾や不一致は、法令の順守や施行に対していかなる拘束力も法的な効力も持ちません。

著作権に関する情報

商標に関する情報

TR-4591-0421-JP