Oracle データベース

Enterprise applications

NetApp
February 10, 2026

目次

Oracle データベース

ONTAP上のOracleデータベース

ONTAPはOracleデータベース向けに設計されています。ONTAPは数十年にわたり、リレーショナルデータベースI/O固有の要求に合わせて最適化されてきました。また、Oracleデータベースのニーズに対応するために、さらにはOracle Inc自体の要求にも対応するために、複数のONTAP機能が開発されました。

本ドキュメントは、これまでに公開されていたテクニカルレポート_TR-3633：『Oracle databases on ONTAP』、TR-4591：『Oracle data protection：Backup、recovery、replication』、TR-4592：『Oracle on MetroCluster』、TR-4534：『Migration of Oracle Databases to NetApp Storage Systems_

ONTAPがデータベース環境にもたらすさまざまな価値に加えて、データベースのサイズ、パフォーマンス要件、データ保護のニーズなど、ユーザのさまざまな要件もあります。NetAppストレージの既知の導入には、VMware ESXで実行される約6、000個のデータベースの仮想環境から、現在996TBのシングルインスタンスデータウェアハウスまで、あらゆるものが含まれます。そのため、NetAppストレージ上にOracleデータベースを設定する際の明確なベストプラクティスはほとんどありません。

NetAppストレージでOracleデータベースを運用するための要件には、2つの方法があります。まず、明確なベストプラクティスが存在する場合は、具体的に説明します。大まかに、Oracleストレージソリューションの設計者が、それぞれのビジネス要件に基づいて対処しなければならない、設計上のさまざまな考慮事項について説明します。

AFF/ FASシステム上のONTAP構成

RAID の場合

RAIDとは、冗長性を使用してドライブの損失からデータを保護することです。

Oracleデータベースやその他のエンタープライズアプリケーションに使用するNetAppストレージの構成では、RAIDレベルに関して疑問が生じることがあります。ストレージアレイ構成に関する従来のOracleのベストプラクティスの多くには、RAIDミラーリングの使用や特定のタイプのRAIDの回避に関する警告が含まれています。これらは有効なポイントを上げますが、これらのソースは、RAID 4およびONTAPで使用されているNetApp RAID DPおよびRAID-TECテクノロジには適用されません。

RAID 4、RAID 5、RAID 6、RAID DP、およびRAID-TECはいずれもパリティを使用して、ドライブ障害によってデータが失われないようにします。これらのRAIDオプションはミラーリングよりもストレージ利用率がはるかに優れていますが、ほとんどのRAID実装には書き込み処理に影響する欠点があります。他のRAID実装で書き込み操作が完了すると、パリティデータを再生成するために複数のドライブ読み取りが必要になる場合があります。これは、一般にRAIDペナルティと呼ばれるプロセスです。

ただし、ONTAPではこのようなRAIDペナルティは発生しません。これは、NetApp WAFL（Write Anywhere File Layout）とRAIDレイヤが統合されているためです。書き込み処理はRAMで結合され、パリティ生成を含む完全なRAIDストライプとして準備されます。ONTAPは書き込みを完了するために読み取りを実行する必要がないため、ONTAPとWAFLはRAIDペナルティを回避できます。レイテンシが重要な処理（Redoロギングなど）のパフォーマンスが妨げられることはありません。また、データファイルのランダム書き込みでは、パリティの再生成が必要になるためにRAIDのペナルティが発生することもありません。

統計的信頼性に関しては、RAID DPでさえRAIDミラーリングよりも優れた保護を提供します。主な問題は、RAIDのリビルド中にドライブに要求が発生することです。ミラーリングされたRAIDセットでは、RAIDセット内のパートナーへのリビルド中にドライブ障害によるデータ損失のリスクが、RAID DPセット内の三重ドライブ障害のリスクよりもはるかに高くなります。

容量管理

データベースやその他のエンタープライズアプリケーションを、予測性と管理性に優れたハイパフォーマンスのエンタープライズストレージで管理するには、データとメタデータを管理するためにドライブ上の空きスペースがいくらか必要です。必要な空きスペースの量は、使用するドライブのタイプやビジネスプロセスによって異なります。

空きスペースとは実際のデータに使用されていないスペースのことで、アグリゲート自体の未割り当てスペースやコンスティチュエントボリューム内の未使用スペースを含みます。シンプロビジョニングも考慮する必要があります。たとえば、あるボリュームに含まれている1TBのLUNのうち、実際のデータに使用されているのは50%だけであるとします。シンプロビジョニング環境では、500GBのスペースが消費されているように見えます。ただし、フルプロビジョニング環境では、1TBの全容量が使用中と表示されます。500GBの未割り当てスペースは非表示になります。このスペースは実際のデータには使用されていないため、空きスペースの合計の計算に含める必要があります。

エンタープライズアプリケーションに使用するストレージシステムに関するNetAppの推奨事項は次のとおりです。

SSDアグリゲート（**AFF**システムを含む）

* NetAppでは*最低10%の空き容量を推奨しています。これには、アグリゲートまたはボリューム内の空きスペース、フルプロビジョニングのために割り当てられているが実際のデータには使用されていない空きスペースなど、すべての未使用スペースが含まれます。論理スペースは重要ではありません。問題は、データストレージに実際に使用できる物理的な空きスペースの量です。

推奨される10%の空きスペースは非常に控えめな値です。SSDアグリゲートでは、パフォーマンスに影響を与えることなく、さらに高い利用率でワークロードをサポートできます。ただし、アグリゲートの使用率が高くなると、使用率を注意深く監視しないと、スペースが不足するリスクも高まります。さらに、容量99%でシステムを実行している場合はパフォーマンスが低下することはありませんが、ハードウェアの追加発注時にシステムが完全にいっぱいにならないように管理作業が必要になる可能性があり、追加のドライブの調達と取り付けに時間がかかることがあります。

HDDアグリゲート（**Flash Pool**アグリゲートを含む）

* NetAppでは*回転式ドライブを使用する場合は、最低15%の空き容量を確保することを推奨しています。これには、アグリゲートまたはボリューム内の空きスペース、フルプロビジョニングのために割り当てられているが実際のデータには使用されていない空きスペースなど、すべての未使用スペースが含まれます。空きスペースが10%に近づくとパフォーマンスが低下します。

Storage Virtual Machine

Oracleデータベースのストレージ管理をStorage Virtual Machine（SVM）で一元化

SVMは、ONTAP CLIではSVMと呼ばれ、ストレージの基本的な機能ユニットであり、VMware ESXサーバ上のゲストと比較すると便利です。

ESXを最初にインストールした時点では、ゲストOSのホストやエンドユーザアプリケーションのサポートなど、事前に設定された機能はありません。仮想マシン（VM）が定義されるまでは空のコンテナです。ONTAPも同様です。ONTAPを最初にインストールした時点では、SVMが作成されるまでデータを提供する機能はありません。データサービスはSVMの特性によって定義されます。

ストレージアーキテクチャの他の要素と同様に、SVMと論理インターフェイス（LIF）の設計に最適なオプションは、拡張要件とビジネスニーズによって大きく異なります。

SVM

ONTAP用にSVMをプロビジョニングする公式のベストプラクティスはありません。適切なアプローチは、管理要件とセキュリティ要件によって異なります。

ほとんどのお客様は、プライマリSVMを1つ運用して日常的な要件のほとんどに対応しつつ、特殊なニーズに対応するSVMを少数作成しています。たとえば、次のようなものを作成できます。

- スペシャリストチームが管理する重要なビジネスデータベース用のSVM
- 開発グループ用のSVM。独自のストレージを個別に管理できるように、完全な管理権限が与えられています。
- 人事や財務報告のデータなど、機密性の高いビジネスデータを格納するSVM。管理チームを限定する必要がある

マルチテナント環境では、各テナントのデータに専用のSVMを割り当てることができます。クラスタ、HAペア、およびノードあたりのSVMとLIFの数の上限は、使用するプロトコル、ノードモデル、およびONTAPのバージョンによって異なります。 を参照してください "NetApp Hardware Universe の略" これらの限界のために。

ONTAP QoSによるパフォーマンス管理

複数のOracleデータベースを安全かつ効率的に管理するには、効果的なQoS戦略が必要です。その理由は、最新のストレージシステムのパフォーマンス機能が絶えず向上していることです。

特に、オールフラッシュストレージの採用が増えたことで、ワークロードの統合が実現しました。回転式メディアに依存するストレージアレイでは、古い回転式ドライブテクノロジではIOPS機能が制限されているため、I/O負荷の高いワークロードの数が限られていました。1つまたは2つの高アクティブデータベースでは、ストレージコントローラが制限に達するずっと前に基盤となるドライブがいっぱいになります。これは変更されました。SSDドライブの数が比較的少ないパフォーマンス機能は、最も強力なストレージコントローラでさえ飽和状態になる可能性があります。つまり、回転式メディアのレイテンシが急激に低下することなく、コントローラのすべての機能を活用できます。

参考例として、シンプルな2ノードHA AFF A800システムでは、レイテンシが1ミリ秒を超える前に最大100万IOPSのランダムIOPSを処理できます。このレベルに達すると予想される単一のワークロードはほとんどありません。このAFF A800システムアレイをフルに活用するには、複数のワークロードをホストする必要がありますが、予測可能性を確保しながら安全にこれを行うには、QoS制御が必要です。

ONTAPには、IOPSと帯域幅の2種類のサービス品質（QoS）があります。QoS制御は、SVM、ボリューム、LUN、およびファイルに適用できます。

IOPS QoS

IOPS QoS制御は、特定のリソースの合計IOPSに基づいていることは明らかですが、IOPS QoSには直感的でない側面がいくつかあります。当初、一部のお客様は、IOPSのしきい値に達したときにレイテンシが明らかに上昇したことに困惑していました。レイテンシの増加は、IOPSが制限されることによる当然の結果です。論理的には、トークンシステムと同様に機能します。たとえば、データファイルが格納されている特定のボリュームに10、000 IOPSの制限がある場合、受信した各I/Oは、処理を続行するために最初にトークンを受信する必要があります。1秒間に10Kを超えるトークンが消費されない限り、遅延は発生しません。IO処理がトークンの受信を待機する必要がある場合、この待機は追加のレイテンシとして表示されます。ワークロードがQoS制限に押し上げられにくくなると、各IOが処理されるまでキューで待機する時間が長くなり、レイテンシが高くなります。

> データベースのトランザクション/ REDOログデータにQoS制御を適用する場合は注意が必要です。通常、Redoログに必要なパフォーマンスはデータファイルよりもはるかに低くなりますが、Redoログのアクティビティはバースト性が高くなります。IOは短時間のパルスで発生し、平均REDO IOレベルに適したQoS制限が実際の要件に対して低すぎる可能性があります。その結果、Redoログのバーストごとに QoSが適用されるため、パフォーマンスが大幅に制限される可能性があります。一般に、REDOログとアーカイブログはQoSによって制限されるべきではありません。

帯域幅QoS

すべてのI/Oサイズが同じではありません。たとえば、あるデータベースが大量の小さなブロック読み取りを実行していて、IOPSのしきい値に達しているとします。 一方、データベースがテーブルのフルスキャン処理を実行している場合もあります。この処理では、大量のブロック読み取りが非常に少なく、大量の帯域幅を消費しますが、IOPSは比較的低くなります。

同様に、VMware環境ではブート時に非常に多くのランダムIOPSが発生する可能性がありますが、外部バックアップ時に実行されるI/Oは少なくても大きくなります。

パフォーマンスを効果的に管理するために、IOPSまたは帯域幅のQoS制限、あるいはその両方が必要になる場合があります。

最小/保証されたQoS

多くのお客様が、QoS保証付きの解決策を求めていますが、これは見た目よりも達成が難しく、無駄になる可能性があります。たとえば、10個のデータベースを10K IOPS保証で配置する場合、10個のデータベースすべてが同時に10K IOPSで実行され、合計で100Kになるようにシステムをサイジングする必要があります。

QoS管理を最小限に抑えるには、重要なワークロードを保護するのが最適です。たとえば、最大IOPSが500Kで、本番ワークロードと開発ワークロードが混在しているONTAPコントローラについて考えてみましょう。特定のデータベースがコントローラを独占しないように、開発ワークロードに最大QoSポリシーを適用する必要があります。次に、最小限のQoSポリシーを本番環境のワークロードに適用して、必要なIOPSを必要なときに常に利用できるようにします。

アダプティブ QoS

アダプティブQoSとは、ONTAPの機能のことで、ストレージオブジェクトの容量に基づいてQoS制限が設定されます。通常、データベースのサイズとそのパフォーマンス要件との間にリンクがないため、データベースで使用されることはほとんどありません。大規模なデータベースはほとんど不活性になる可能性がありますが、小規模なデータベースはIOPS負荷が最も高くなる可能性があります。

アダプティブQoSは、仮想化データストアで非常に役立ちます。このようなデータセットのIOPS要件は、デ

ータベースの合計サイズに相関する傾向があるためです。1TBのVMDKファイルを格納した新しいデータストアでは、2TBデータストアの約半分のパフォーマンスが必要になる可能性があります。アダプティブQoSを使用すると、データストアにデータが入力されたときに、QoS制限を自動的に増やすことができます。

効率性

ONTAPのスペース効率化機能は、Oracleデータベース向けに最適化されています。ほとんどの場合、すべての効率化機能を有効にした状態でデフォルトのままにすることを推奨します。

圧縮、コンパクション、重複排除などのスペース効率化機能は、特定の量の物理ストレージに収まる論理データの量を増やすように設計されています。その結果、コストと管理オーバーヘッドが削減されます。

圧縮とは、大まかに言って、データのパターンを検出してスペースを削減する方法でエンコードする数学的プロセスです。一方、重複排除機能は、実際に繰り返されるデータブロックを検出し、不要なコピーを削除します。コンパクションを使用すると、複数の論理ブロックのデータをメディア上の同じ物理ブロックで共有できます。

Storage Efficiencyとフラクショナルリザベーションの連動については、シンプロビジョニングに関する以下のセクションを参照してください。

圧縮

オールフラッシュストレージシステムが登場する以前は、アレイベースの圧縮の価値は限られていました。I/O負荷の高いワークロードのほとんどでは、許容可能なパフォーマンスを提供するために非常に多数のスピンドルが必要だったためです。ストレージシステムには、ドライブ数が多いことの副作用として、必要以上の容量が常に搭載されていました。この状況は、ソリッドステートストレージの登場によって変化しました。優れたパフォーマンスを得るためだけにドライブを過剰にオーバープロビジョニングする必要はもうありません。ストレージシステムのドライブスペースは、実際の容量ニーズに合わせて調整できます。

ソリッドステートドライブ（SSD）ではIOPSが向上するため、ほとんどの場合、回転式ドライブに比べてコストを削減できますが、圧縮を使用すると、ソリッドステートメディアの実効容量を増やすことで、さらに削減効果を高めることができます。

データを圧縮する方法はいくつかあります。多くのデータベースには独自の圧縮機能が搭載されていますが、お客様の環境ではこのような圧縮機能はほとんど見られません。その理由は、通常、圧縮データを*変更*するとパフォーマンスが低下することに加え、一部のアプリケーションではデータベースレベルの圧縮のライセンスコストが高くなることにあります。最後に、データベース処理のパフォーマンスが全体的に低下します。実際のデータベース作業ではなく、データの圧縮と解凍を実行するCPUに、高いCPU単位のライセンスコストを支払うことはほとんど意味がありません。より適切な方法は、圧縮処理をストレージシステムにオフロードすることです。

適応圧縮

アダプティブ圧縮は、レイテンシがマイクロ秒単位で測定されるオールフラッシュ環境であっても、パフォーマンスに影響を及ぼさないエンタープライズワークロードで徹底的にテストされています。一部のお客様から、圧縮機能によってデータがキャッシュ内で圧縮されたままになるため、パフォーマンスが向上したとの報告もあります。これは、コントローラで使用可能なキャッシュ容量が実質的に増加するためです。

ONTAPは物理ブロックを4KB単位で管理します。アダプティブ圧縮では、デフォルトの圧縮ブロックサイズである8KBが使用されます。つまり、データは8KB単位で圧縮されます。これは、リレーショナルデータベースで最もよく使用される8KBのブロックサイズに一致します。圧縮アルゴリズムは、より多くのデータが1つ

の単位として圧縮されるので、より効率的になります。圧縮ブロックサイズが32KBの場合、8KBの圧縮ブロックユニットよりもスペース効率に優れています。つまり、デフォルトの8KBのブロックサイズを使用するアダプティブ圧縮の場合、削減率はわずかに低くなりますが、圧縮ブロックサイズを小さくすることには大きなメリットがあります。データベースワークロードには、大量の上書きアクティビティが含まれています。32KBの圧縮されたデータブロックの8KBを上書きするには、32KBの論理データ全体を読み取って解凍し、必要な8KB領域を更新してから再度圧縮し、32KB全体をドライブに書き込む必要があります。この処理はストレージシステムでは非常にコストがかかります。このため、圧縮ブロックサイズの大きい競合ストレージアレイでも、データベースワークロードのパフォーマンスが大幅に低下します。

適応圧縮で使用されるブロックサイズは、最大32KBまで拡張できます。これによりストレージ効率が向上する可能性があります。このようなデータがアレイに大量に格納されている場合は、トランザクションログやバックアップファイルなどの静止ファイルについて検討する必要があります。状況によっては、適応圧縮のブロックサイズをそれに合わせて増やすことで、16KBまたは32KBのブロックサイズを使用するアクティブデータベースでもメリットが得られる場合があります。この方法がお客様のワークロードに適しているかどうかについては、NetAppまたはパートナーの担当者にお問い合わせください。

ストリーミングバックアップデスティネーションでは、重複排除と一緒に8KBを超える圧縮ブロックサイズを使用しないでください。これは、バックアップデータへのわずかな変更が32KBの圧縮ウィンドウに影響するためです。ウィンドウが移動すると、圧縮されたデータはファイル全体で異なります。重複排除は圧縮後に実行されます。つまり、重複排除エンジンは、圧縮された各バックアップを別 々 に認識します。ストリーミングバックアップの重複排除が必要な場合は、8KBのブロックアダプティブ圧縮のみを使用します。アダプティブ圧縮を使用することを推奨します。アダプティブ圧縮はブロックサイズが小さく、重複排除による効率化の妨げにならないためです。同様の理由から、ホスト側の圧縮も重複排除による効率化の妨げになります。

圧縮のアライメント

データベース環境でアダプティブ圧縮を使用する場合は、圧縮ブロックのアライメントについて考慮する必要があります。これは、非常に特定のブロックでランダムオーバーライトが発生するデータについてのみ考慮する必要があります。このアプローチは、ファイルシステム全体のアライメントと概念的に似ています。ファイルシステムの開始は4Kデバイスの境界に合わせて調整する必要があり、ファイルシステムのブロックサイズは4Kの倍数でなければなりません。

たとえば、ファイルへの8KBの書き込みは、ファイルシステム自体の8KBの境界にアライメントされている場合にのみ圧縮されます。これは、ファイルの最初の8KB、ファイルの2番目の8KBなどに配置する必要があることを意味します。アライメントを正しく行う最も簡単な方法は、正しいLUNタイプを使用することです。作成するパーティションには、デバイスの先頭から8Kの倍数のオフセットを設定し、データベースのブロックサイズの倍数のファイルシステムのブロックサイズを使用する必要があります。

バックアップやトランザクションログなどのデータは、複数のブロックにまたがるシーケンシャル書き込み処理であり、すべて圧縮されます。したがって、アライメントを考慮する必要はありません。問題となるI/Oパターンは、ファイルのランダムオーバーライトだけです。

データコンパクション

データコンパクションは、圧縮効率を向上させるテクノロジです。前述したように、アダプティブ圧縮では4KBのWAFLブロックに8KBのI/Oが格納されるため、削減率は最大でも2：1です。ブロックサイズが大きい圧縮方式では、効率性が向上します。ただし、小さなブロックの上書きが発生するデータには適していません。32KBのデータユニットを解凍して8KB部分を更新し、再度圧縮してからドライブにライトバックすると、オーバーヘッドが発生します。

データコンパクションでは、複数の論理ブロックを物理ブロック内に格納できます。たとえば、テキストブロックや部分的にフルブロックなど、圧縮率の高いデータを含むデータベースは、8KBから1KBに圧縮できます。コンパクションを使用しない場合、この1KBのデータが4KBブロック全体を占有します。インラインデータコンパクションでは、圧縮された1KBのデータを、他の圧縮データと一緒にわずか1KBの物理スペースに格納できます。これは圧縮テクノロジではありません。ドライブのスペースをより効率的に割り当てる方法なので、検出できるほどのパフォーマンスへの影響はありません。

得られる削減効果の程度はさまざまです。すでに圧縮または暗号化されているデータは、通常それ以上圧縮することはできないため、コンパクションによるメリットはありません。一方、初期化されたばかりのデータファイルで、ブロックメタデータとゼロブロックしか含まれていない場合は、最大80：1まで圧縮できます。

温度に基づくストレージ効率

Temperature Sensitive Storage Efficiency（TSSE）は、ONTAP 9 .8以降で使用できます。ブロックアクセスのヒートマップを使用して、アクセス頻度の低いブロックを特定し、より効率的に圧縮します。

重複排除

重複排除とは、データセットから重複するブロックサイズを削除することです。たとえば、同じ4KBブロックが10個のファイルに存在する場合、重複排除機能は、10個のファイルすべてのうち、その4KBブロックを同じ4KBの物理ブロックにリダイレクトします。その結果、そのデータの効率が10分の1に向上します。

VMwareゲストブートLUNなどのデータは、同じオペレーティングシステムファイルの複数のコピーで構成されるため、通常は重複排除が非常に効果的です。100:1以上の効率が観測されている。

一部のデータに重複データが含まれていません。たとえば、Oracleブロックには、データベースに対してグローバルに一意のヘッダーと、ほぼ一意のトレーラが含まれています。そのため、Oracleデータベースの重複排除によって1%以上の削減効果が得られることはほとんどありません。MS SQLデータベースでの重複排除はやや優れていますが、ブロックレベルでの固有のメタデータは依然として制限されています。

16KBでブロックサイズが大きいデータベースでは、最大15%のスペース削減効果が確認されたケースがいくつかあります。各ブロックの最初の4KBにはグローバルに一意なヘッダーが含まれ、最後の4KBブロックにはほぼ一意のトレーラが含まれます。内部ブロックは重複排除の対象となりますが、実際には、初期化されたデータの重複排除にほぼ完全に起因しています。

競合するアレイの多くは、データベースが複数回コピーされていると仮定して、データベースの重複排除機能があると主張しています。この点では、NetAppの重複排除も使用できますが、ONTAPにはNetApp FlexCloneテクノロジというより優れたオプションがあります。最終的な結果は同じで、基盤となる物理ブロックの大部分を共有するデータベースのコピーが複数作成されます。FlexCloneを使用すると、時間をかけてデータベースファイルをコピーしてから重複を排除するよりも、はるかに効率的です。重複は最初から作成されないため、実際には重複排除ではなく重複排除です。

効率性とシンプロビジョニング

効率化機能はシンプロビジョニングの一形態です。たとえば、100GBのボリュームを使用している100GBのLUNを50GBに圧縮するとします。ボリュームが100GBのままなので、実際の削減はまだ実現されていません。削減されたスペースをシステムの他の場所で使用できるように、まずボリュームのサイズを縮小する必要があります。100GBのLUNにあとから変更した結果、データの圧縮率が低下すると、LUNのサイズが大きくなり、ボリュームがいっぱいになる可能性があります。

シンプロビジョニングは、管理を簡易化しながら、使用可能な容量を大幅に改善し、コストを削減できるため、強く推奨されます。これは、単純なデータベース環境では、多くの場合、空のスペース、多数のボリュームやLUN、圧縮可能なデータが含まれているためです。シックプロビジョニングでは、ボリュームとLUNのス

トレージにスペースがリザーブされます。これは、100%フルになり、100%圧縮不可能なデータが含まれる場合に限られます。これは起こりそうもないことですシンプロビジョニングを使用すると、スペースを他の場所で再生して使用できます。また、容量の管理は、多数の小さいボリュームやLUNではなく、ストレージシステム自体に基づいて行うことができます。

一部のお客様は、特定のワークロードにシックプロビジョニングを使用するか、一般的には確立された運用と調達の手法に基づいてシックプロビジョニングを使用します。

> ボリュームがシックプロビジョニングされている場合は、コマンドを使用した解凍や重複排除の削除など、そのボリュームのすべての効率化機能を完全に無効にするように注意する必要があります `sis undo`。ボリュームは出力に表示されません `volume efficiency show`。有効になっている場合、ボリュームはまだ部分的に効率化機能用に設定されています。その結果、オーバーライトギャランティの動作が異なります。そのため、設定がオーバーサイトされるとボリュームが予期せずスペース不足になり、データベースI/Oエラーが発生する可能性が高くなります。

効率化のベストプラクティス

- NetAppの推奨事項*：

AFFのデフォルト

オールフラッシュAFFシステムで実行されているONTAPで作成されたボリュームは、すべてのインライン効率化機能が有効になった状態でシンプロビジョニングされます。一般にデータベースには重複排除機能はなく、圧縮不可能なデータも含まれている可能性がありますが、デフォルト設定はほとんどすべてのワークロードに適しています。ONTAPは、あらゆる種類のデータとI/Oパターンを効率的に処理するように設計されており、削減効果があるかどうかは関係ありません。デフォルトは、理由が完全に理解されていて、逸脱するメリットがある場合にのみ変更する必要があります。

一般的な推奨事項

- ボリュームやLUNがシンプロビジョニングされていない場合は、すべての効率化設定を無効にする必要があります。これらの機能を使用しても削減は得られず、シックプロビジョニングとスペース効率化が有効になっていると、スペース不足エラーなどの予期しない動作が発生する可能性があるためです。
- バックアップやデータベーストランザクションログなどでデータが上書きされない場合は、クーリング期間を短くしてTSSEを有効にすることで、効率を高めることができます。
- アプリケーションレベルで圧縮がすでに有効になっているファイルが暗号化されている場合など、一部のファイルには圧縮不可能なデータが大量に含まれていることがあります。上記のいずれかに該当する場合は、圧縮可能なデータを含む他のボリュームでより効率的に処理できるように、圧縮を無効にすることを検討してください。
- データベースバックアップでは、32KBの圧縮機能と重複排除機能の両方を使用しないでください。を参照してください [適応圧縮] を参照してください。

シンプロビジョニング

Oracleデータベースのシンプロビジョニングでは、ストレージシステムに物理的に使用可能なスペースよりも多くのスペースが設定されるため、慎重な計画が必要になります。適切に行うと、大幅なコスト削減と管理性の向上につながるため、努力する価値は非常に高くなります。

シンプロビジョニングにはさまざまな形式があり、ONTAPがエンタープライズアプリケーション環境に提供する多くの機能に欠かせない機能です。シンプロビジョニングも効率化テクノロジと密接に関連しています。効率化機能を使用すると、ストレージシステムに実際に存在するよりも多くの論理データを格納できます。

Snapshotを使用する場合、シンプロビジョニングが必要になります。たとえば、NetAppストレージ上の一般的な10TBのデータベースには、約30日間のSnapshotが含まれています。この構成では、アクティブファイルシステムに表示されるデータは約10TB、Snapshot専用のデータは300TBになります。合計310TBのストレージは、通常、約12$_{15TB}$のスペースに配置されます。アクティブデータベースは$_{10TB}$を消費しますが、残りの$_{300TB}$のデータは元のデータに加えられた変更のみが格納されるため、$_{2TB}$5TBのスペースしか必要としません。

クローニングもシンプロビジョニングの一例です。NetAppの主要なお客様が、80TBのデータベースのクローンを40個作成し、開発用に使用しました。これらのクローンを使用している40人の開発者全員がすべてのデータファイルのすべてのブロックを上書きした場合、3.2PBを超えるストレージが必要になります。実際には書き替え率は低く、変更のみがドライブに格納されるため、必要なスペースは合計で40TBに近くなります。

スペース管理

データの変更率が予期せず増加する可能性があるため、アプリケーション環境のシンプロビジョニングには注意が必要です。たとえば、データベーステーブルのインデックスを再作成したり、VMwareゲストに大規模なパッチを適用したりすると、Snapshotによるスペース消費が急増します。バックアップの配置が間違っていると、非常に短時間で大量のデータが書き込まれる可能性があります。最後に、ファイルシステムの空きスペースが予期せず不足した場合、一部のアプリケーションのリカバリが困難になることがあります。

幸いなことに、これらのリスクは慎重に構成することで対処できます。 `volume-autogrow` および `snapshot-autodelete` ポリシー：名前からわかるように、これらのオプションを使用すると、Snapshotによって消費されるスペースを自動的にクリアしたり、追加データに対応するためにボリュームを拡張したりするポリシーを作成できます。多くのオプションが用意されており、ニーズはお客様によって異なります。

を参照してください "論理ストレージ管理に関する文書" を参照してください。

フラクショナルリザベーション

フラクショナルリザーブは、ボリューム内でのスペース効率に関するLUNの動作です。オプション `fractional-reserve` が100%に設定されている場合、ボリュームのスペースを使い切ることなく、任意のデータパターンでボリューム内のすべてのデータを100%書き替えることができます。

たとえば、1TBのボリュームに配置された単一の250GB LUNにデータベースが格納されているとします。Snapshotを作成すると、ただちにボリュームに250GBのスペースが追加でリザーブされ、何らかの理由でボリュームのスペースが不足することはありません。データベースボリュームのすべてのバイトの上書きが必要になることはほとんどないため、フラクショナルリザーブの使用は一般に無駄です。決して発生しないイベントのためにスペースを予約する理由はありません。ただし、ストレージシステムのスペース消費を監視できず、スペースが不足しないように保証する必要がある場合は、Snapshotの使用に100%のフラクショナルリザベーションが必要になります。

圧縮と重複排除

圧縮と重複排除はどちらもシンプロビジョニングの形式です。たとえば、50TBのデータ容量を30TBに圧縮すると、20TBが削減されます。圧縮によってメリットが得られるようにするには、20TBの一部を他のデータに使用するか、50TB未満のストレージシステムを購入する必要があります。その結果、ストレージシステムで実際に使用可能な量よりも多くのデータが格納されます。データの観点から見ると、ドライブでは30TBしか占有していないにもかかわらず、50TBのデータがあります。

データセットの圧縮率は常に変化し、実際のスペースの消費量が増加する可能性があります。このように消費

量が増加するため、他の形式のシンプロビジョニングと同様に、監視と使用の観点から圧縮を管理する必要があります。 `volume-autogrow` および `snapshot-autodelete`。

圧縮機能と重複排除機能の詳細については、efficiency.htmlのリンクを参照してください。

圧縮とフラクショナルリザベーション

圧縮はシンプロビジョニングの一形態です。フラクショナルリザベーションは圧縮の使用に影響します。スペースはSnapshotの作成前にリザーブされる点に注意してください。通常、フラクショナルリザーブが重要になるのはSnapshotが存在する場合のみです。Snapshotがない場合、フラクショナルリザーブは重要ではありません。これは、圧縮の場合には当てはまりません。圧縮が有効なボリュームでLUNを作成すると、ONTAPではSnapshotに対応するためのスペースが確保されます。この動作は設定時に混乱を招く可能性がありますが、これは想定される動作です。

たとえば、10GBのボリュームに5GBのLUNが格納され、2.5GBに圧縮されてSnapshotが作成されていないとします。次の2つのシナリオを検討します。

- フラクショナルリザーブ= 100では7.5GBが使用されます。
- フラクショナルリザーブ= 0の場合、2.5GBの使用率が得られます。

最初のシナリオでは、現在のデータ用に2.5GBのスペースが使用され、スナップショットの使用を想定してソースの100%の切り替えに使用される5GBのスペースが使用されます。2番目のシナリオでは、追加のスペースは確保されません。

この状況は混乱しているように見えるかもしれませんが、実際に遭遇することはほとんどありません。圧縮はシンプロビジョニングを意味し、LUN環境でのシンプロビジョニングにはフラクショナルリザベーションが必要です。圧縮されたデータは圧縮不可能なデータで上書きされる可能性があります。つまり、圧縮によって削減効果が得られるように、ボリュームをシンプロビジョニングする必要があります。

- NetAppでは*次の予約構成を推奨しています。
- 設定 `fractional-reserve` 基本的な容量監視と `volume-autogrow` および `snapshot-autodelete`。
- 設定 `fractional-reserve` 監視機能がない場合、または何らかの状況でスペースを使い切ることができない場合は、100にします。

空きスペースと**LVM**スペースの割り当て

ファイル システム環境におけるアクティブ LUN のシン プロビジョニングの効率は、データが削除されるにつれて時間の経過とともに失われる可能性があります。削除されたデータがゼロで上書きされるか (link:oracle-storage-san-config-asm ru.html[ASMRU] も参照)、TRIM/UNMAP スペース再利用によってスペースが解放されない限り、「消去された」データはファイル システム内の未割り当ての空白をどんどん占有することになります。さらに、データファイルは作成時にフルサイズに初期化されるため、アクティブ LUN のシン プロビジョニングの使用は多くのデータベース環境で制限されます。

LVMの構成を慎重に計画すると、効率が向上し、ストレージのプロビジョニングやLUNのサイズ変更の必要性を最小限に抑えることができます。Veritas VxVMやOracle ASMなどのLVMを使用すると、基盤となるLUNが複数のエクステントに分割され、必要な場合にのみ使用されます。たとえば、最初は2TBのサイズだったデータセットが、やがて10TBに拡張される可能性がある場合、このデータセットを、シンプロビジョニングされた10TBのLUNがLVMディスクグループにまとめられて配置できます。作成時に消費されるスペースは2TBにすぎず、データ量の増大に対応するためにエクステントが割り当てられた場合にのみ追加のスペースが必要になります。このプロセスは、スペースが監視されているかぎり安全です。

ONTAPフェイルオーバー/スイッチオーバー

Oracleデータベースの処理が中断されないようにするには、ストレージのテイクオーバーとスイッチオーバーの機能を理解しておく必要があります。また、テイクオーバー処理やスイッチオーバー処理で使用される引数を正しく使用しないと、データの整合性に影響する可能性があります。

- 通常の状態では、特定のコントローラへの書き込みは、パートナーに同期ミラーリングされます。NetApp MetroCluster環境では、書き込みはリモートコントローラにもミラーリングされます。書き込みがすべての場所の不揮発性メディアに格納されるまで、ホストアプリケーションに確認応答は返されません。
- 書き込みデータを格納するメディアは、不揮発性メモリ（NVMEM）と呼ばれます。不揮発性ランダムアクセスメモリ（NVRAM）と呼ばれることもあります。機能はジャーナルですが、書き込みキャッシュとみなすことができます。通常の処理でNVMEMのデータが読み取られることはなく、ソフトウェアまたはハードウェアに障害が発生した場合のデータ保護にのみ使用されます。ドライブにデータが書き込まれると、NVMEMではなくシステムのRAMからデータが転送されます。
- テイクオーバー処理では、ハイアベイラビリティ（HA）ペアの一方のノードがパートナーの処理をテイクオーバーします。スイッチオーバーは基本的に同じですが、IT環境 MetroCluster構成ではリモートノードがローカルノードの機能を引き継ぎます。

定期的なメンテナンス作業中は、ネットワークパスの変更によって一時的に運用が停止する可能性がある場合を除き、ストレージのテイクオーバーやスイッチオーバーは透過的に行われる必要があります。ただし、ネットワークの設定は複雑なため、エラーが発生しやすいため、NetAppでは、ストレージシステムを本番環境に移行する前に、テイクオーバーとスイッチオーバーの処理を徹底的にテストすることを強く推奨します。これは、すべてのネットワークパスが正しく設定されていることを確認する唯一の方法です。SAN環境では、コマンドの出力を慎重に確認します。 `sanlun lun show -p` 想定されるすべてのプライマリパスとセカンダリパスが使用可能であることを確認します。

テイクオーバーやスイッチオーバーを強制的に実行する場合は注意が必要です。これらのオプションを使用してストレージ構成を強制的に変更すると、ドライブを所有するコントローラの状態が無視され、代わりのノードが強制的にドライブの制御を引き継ぐことになります。テイクオーバーの強制が正しく行われないと、データの損失や破損が発生する可能性があります。これは、強制的なテイクオーバーやスイッチオーバーによってNVMEMの内容が破棄される可能性があるためです。テイクオーバーまたはスイッチオーバーの完了後にそのデータが失われると、データベースから見ると、ドライブに格納されているデータが少し古い状態に戻る可能性があります。

通常のHAペアを使用した強制テイクオーバーが必要になることはほとんどありません。ほぼすべての障害シナリオでは、ノードがシャットダウンし、パートナーに通知して自動フェイルオーバーが実行されます。一部のエッジケース（ローリング障害など）では、ノード間のインターコネクトが失われたあとに一方のコントローラが失われた場合など、強制テイクオーバーが必要になります。この場合、コントローラで障害が発生する前にノード間のミラーリングが失われるため、稼働しているコントローラには実行中の書き込みのコピーが存在しなくなります。その後、テイクオーバーを強制的に実行する必要があります。つまり、データが失われる可能性があります。

同じ論理環境A MetroClusterスイッチオーバー。通常の状態では、スイッチオーバーはほぼ透過的に実行されます。ただし、災害が発生すると、サバイバーサイトとディザスタサイトの間の接続が失われる可能性があります。サバイバーサイトから見ると、問題はサイト間の接続の中断にすぎず、元のサイトが引き続きデータを処理している可能性があります。ノードがプライマリコントローラの状態を検証できない場合は、強制スイッチオーバーのみが実行できます。

- NetAppでは、次の注意事項を遵守することを推奨しています。
- テイクオーバーやスイッチオーバーを誤って強制的に実行しないように十分注意してください。通常は強制は必要ありません。強制的に変更すると、原因のデータが失われる可能性があります。
- テイクオーバーやスイッチオーバーの強制実行が必要な場合は、アプリケーションがシャットダウンされ、すべてのファイルシステムがディスマウントされ、論理ボリュームマネージャ（LVM）ボリュームグループが分離されていることを確認してください。ASMディスクグループをアンマウントする必要があります。
- MetroClusterの強制スイッチオーバーが発生した場合は、障害が発生したノードを残りのすべてのストレージリソースからフェンシングします。詳細については、該当するバージョンのONTAPの『MetroCluster管理およびディザスタリカバリガイド』を参照してください。

MetroClusterと複数のアグリゲート

MetroClusterは同期レプリケーションテクノロジで、接続が中断されると非同期モードに切り替わります。これはお客様からの最も一般的な要求です。同期レプリケーションが保証されていると、サイト接続が中断されるとデータベースI/Oが完全に停止し、データベースのサービスが停止します。

MetroClusterを使用すると、接続がリストアされたあとにアグリゲートが迅速に再同期されます。他のストレージテクノロジとは異なり、MetroClusterでは、サイト障害後に完全な再ミラーリングを行う必要はありません。変更された差分のみを出荷する必要があります。

複数のアグリゲートにまたがるデータセットでは、ローリングディザスタシナリオでデータリカバリ手順を追加する必要があるというわずかなリスクがあります。具体的には、（a）サイト間の接続が中断された場合、（b）接続がリストアされた場合、（c）アグリゲートが同期されている状態と同期されていない状態になった場合、 そして（d）プライマリサイトが失われると、サバイバーサイトが作成され、アグリゲートが相互に同期されなくなります。この場合、データセットの一部が相互に同期され、アプリケーション、データベース、またはデータストアをリカバリなしで起動することはできません。データセットが複数のアグリゲートにまたがっている場合、NetAppでは、Snapshotベースのバックアップと多数のツールのいずれかを活用して、このような異常な状況で迅速にリカバリできるかどうかを検証することを強く推奨しています。

ASA r2 システムでのONTAP構成

RAID の場合

RAID とは、ドライブ障害からデータを保護するためにパリティベースの冗長性を使用することを指します。ASA r2 は、 AFFおよびFASシステムと同じONTAP RAID テクノロジーを使用して、複数のディスク障害に対する堅牢な保護を保証します。

ONTAP はASA r2 システムの RAID 構成を自動的に実行します。これは、 ASA r2 パーソナリティで導入された簡素化されたストレージ管理エクスペリエンスの中核コンポーネントです。

ASA r2 での自動 RAID 構成に関する主な詳細は次のとおりです。

- ストレージ アベイラビリティ ゾーン (SAZ): 従来の集約および RAID グループを手動で管理する代わりに、 ASA r2 はストレージ アベイラビリティ ゾーン (SAZ) を使用します。これらは、HA ペア用の共有の RAID 保護されたディスク プールであり、両方のノードが同じストレージに完全にアクセスできます。
- 自動配置: ストレージ ユニット (LUN または NVMe 名前空間) が作成されると、 ONTAP はSAZ 内にボリ

ュームを自動的に作成し、最適なパフォーマンスと容量のバランスになるように配置します。

- 手動のアグリゲート管理なし: 従来のアグリゲートおよび RAID グループ管理コマンドはASA r2 ではサポートされていません。これにより、管理者が RAID グループのサイズ、パリティ ディスク、またはノードの割り当てを手動で計画する必要がなくなります。
- 簡素化されたプロビジョニング: プロビジョニングは、基盤となる物理 RAID レイアウトではなく、ストレージ ユニットに重点を置いた System Manager または簡素化された CLI コマンドによって処理されます。
- ワークロードの再バランス調整: 2025 リリース (ONTAP 9.17.1) 以降、 ONTAP はHA ペアのノード間でワークロードを自動的に再バランス調整し、手動による介入なしにパフォーマンスとスペース使用率のバランスが維持されるようにします。

ASA r2 は、ONTAP のデフォルトの RAID テクノロジー (ほとんどの構成ではRAID DP 、非常に大規模な SSD プールではRAID-TEC)を自動的に使用します。これにより、手動で RAID を選択する必要がなくなります。これらのパリティベースの RAID レベルは、ミラーリングよりも優れたストレージ効率と信頼性を提供します。ミラーリングは、古い Oracle のベスト プラクティスで推奨されることが多いですが、 ASA r2 には関係ありません。ONTAP は、 WAFL統合によって従来の RAID 書き込みペナルティを回避し、REDO ログやランダム データファイル書き込みなどの Oracle ワークロードに最適なパフォーマンスを保証します。自動化された RAID 管理とストレージ可用性ゾーンを組み合わせることで、 ASA r2 は Oracle データベースに高可用性とエンタープライズクラスの保護を提供します。

容量管理

データベースやその他のエンタープライズアプリケーションを、予測性と管理性に優れたハイパフォーマンスのエンタープライズストレージで管理するには、データとメタデータを管理するためにドライブ上の空きスペースがいくらか必要です。必要な空きスペースの量は、使用するドライブのタイプやビジネスプロセスによって異なります。

ASA r2 は集約の代わりにストレージ可用性ゾーン (SAZ) を使用しますが、原則は同じです。空き領域には、実際のデータ、スナップショット、またはシステム オーバーヘッドによって消費されない物理容量が含まれます。シン プロビジョニングも考慮する必要があります。論理割り当てでは実際の物理的な使用量が反映されません。

エンタープライズ アプリケーションに使用されるASA r2 ストレージ システムに対するNetApp の推奨事項は次のとおりです。

ASA r2 システムの **SSD** プール

* NetApp は* ASA r2 環境で少なくとも 10% の物理空き領域を維持することを推奨します。このガイドラインは、 ASA r2 システムで使用される SSD 専用プールに適用され、SAZ およびストレージ ユニット内のすべての未使用スペースが含まれます。論理スペースは重要ではありません。データの保存に使用できる実際の空き物理スペースに焦点が当てられます。

ASA r2 はパフォーマンスを低下させることなく高い使用率を維持できますが、最大容量に近い状態で動作させると、ストレージを拡張する際にスペース不足や管理オーバーヘッドのリスクが増大します。90% を超える使用率で実行してもパフォーマンスには影響しませんが、管理が複雑になり、追加ドライブのプロビジョニングが遅れる可能性があります。

ASA r2 システムは、最大 128 TB のストレージ ユニットと HA ペアあたり最大 2 PB の SAZ サイズをサポートし、 ONTAP はノード間で容量を自動的に分散します。スナップショット、シンプロビジョニングワークロード、および将来の成長のために十分な空き領域を確保するには、クラスター、SAZ、およびストレージ ユ

ニット レベルでの使用率を監視することが不可欠です。容量が重大なしきい値（使用率約 90%）に近づくと、パフォーマンスと回復力を維持するために、追加の SSD をグループ（最低 6 台のドライブ）で追加する必要があります。

Storage Virtual Machine

ASA r2 システム上の Oracle データベース ストレージ管理も、 ONTAP CLI では vserver と呼ばれるストレージ仮想マシン (SVM) に集中化されています。

SVM は、VMware ESXサーバ上のゲスト VM と同様に、 ONTAPにおけるストレージ プロビジョニングとセキュリティの基本単位です。ONTAPがASA r2 に初めてインストールされると、SVM が作成されるまでデータ提供機能はありません。SVM は、SAN 環境のパーソナリティとデータ サービスを定義します。

ASA r2 システムは、ブロック プロトコル (FC、iSCSI、NVMe/FC、NVMe/TCP) をサポートするように合理化され、NAS 関連の機能が削除された SAN 専用のONTAPパーソナリティを使用します。これにより管理が簡素化され、すべての SVM 構成が SAN ワークロードに合わせて最適化されます。AFF/ FASシステムとは異なり、 ASA r2 では、ホーム ディレクトリや NFS 共有などの NAS サービスのオプションは公開されません。

クラスタが作成されると、 ASA r2 は SAN プロトコルが有効になっている svm1 という名前のデフォルトのデータ SVM を自動的にプロビジョニングします。この SVM は、プロトコル サービスを手動で構成する必要なく、ブロック ストレージ操作に対応しています。デフォルトでは、この SVM の IP データ LIF は iSCSI および NVMe/TCP プロトコルをサポートし、default-data-blocks サービス ポリシーを使用するため、SAN ワークロードの初期セットアップが簡素化されます。管理者は、後でパフォーマンス、セキュリティ、またはマルチテナントの要件に基づいて追加の SVM を作成したり、LIF 構成をカスタマイズしたりできます。

> SAN プロトコルの論理インターフェイス (LIF) は、パフォーマンスと可用性の要件に基づいて設計する必要があります。ASA r2 は iSCSI、FC、および NVMe LIF をサポートしていますが、 ASA r2 は NVMe および SCSI ホストに共有ネットワークを使用するため、自動 iSCSI LIF フェイルオーバーはデフォルトでは有効になっていないことに注意してください。自動フェイルオーバーを有効にするには、 "iSCSI専用LIF"。

SVM

他のONTAPプラットフォームと同様に、作成する SVM の数に関する公式のベスト プラクティスはなく、管理とセキュリティの要件に応じて決定されます。

ほとんどのお客様は、日常業務用に単一のプライマリ SVM を運用し、次のような特別なニーズに合わせて追加の SVM を作成します。

- 専門チームによって管理される重要なビジネスデータベース専用のSVM
- 委任された管理制御を持つ開発グループ用の SVM
- 制限された管理アクセスを必要とする機密データ用の SVM

マルチテナント環境では、各テナントに専用の SVM を割り当てることができます。クラスタ、HA ペア、ノードあたりの SVM および LIF の数の制限は、使用されているプロトコル、ノード モデル、およびONTAPのバージョンによって異なります。ご相談ください "NetApp Hardware Universe の略" これらの制限については。

> ASA r2 は、 ONTAP 9.18.1 以降、クラスタあたりおよび HA ペアあたり最大 256 個の SVM をサポートします (以前のリリースでは 32 個でした)。

ASA r2 システムでのONTAP QoS によるパフォーマンス管理

ASA r2 上で複数の Oracle データベースを安全かつ効率的に管理するには、効果的な QoS 戦略が必要です。ASA r2 システムは、極めて高いパフォーマンスとワークロード統合のために設計されたオールフラッシュSAN プラットフォームであるため、これが特に重要です。

比較的少数の SSD では、最も強力なコントローラでも飽和状態になる可能性があるため、複数のワークロードにわたって予測可能なパフォーマンスを確保するには QoS 制御が不可欠です。参考までに、 ASA A1K や A90 などのASA r2 システムは、数十万から 100 万を超える IOPS をミリ秒未満の遅延で提供できます。このパフォーマンス レベルを消費する単一のワークロードはほとんどないため、完全に利用するには、通常、複数のデータベースまたはアプリケーションをホストする必要があります。これを安全に行うには、リソースの競合を防ぐための QoS ポリシーが必要です。

ASA r2 上のONTAP QoS は、 AFF/ FASシステムと同じように動作し、IOPS と帯域幅という 2 つの主要なタイプの制御を備えています。QoS 制御は SVM および LUN に適用できます。

IOPS QoS

IOPS ベースの QoS は、特定のリソースの合計 IOPS を制限します。ASA r2 では、QoS ポリシーを SVM レベルおよび LUN などの個々のストレージ オブジェクトに適用できます。ワークロードが IOPS 制限に達すると、追加の I/O 要求がトークンのキューに入れられ、遅延が発生します。これは想定された動作であり、単一のワークロードがシステム リソースを独占することを防ぎます。

> データベース トランザクション/REDO ログ データに QoS 制御を適用する場合は注意が必要です。これらのワークロードはバースト性が高く、平均的なアクティビティに対しては妥当と思われる QoS 制限が、ピーク時のバーストに対しては低すぎる可能性があり、深刻なパフォーマンスの問題を引き起こします。一般に、再実行およびアーカイブ ログは QoS によって制限されるべきではありません。

帯域幅QoS

帯域幅ベースの QoS は、スループットを Mbps 単位で制限します。これは、完全なテーブルスキャンやバックアップ操作など、大量の帯域幅を消費するが IOPS は比較的少ない、大規模なブロックの読み取りや書き込みを実行するワークロードに役立ちます。IOPS と帯域幅の制限を組み合わせることで、よりきめ細かな制御が可能になります。

最小/保証されたQoS

最小 QoS ポリシーは、重要なワークロードのパフォーマンスを確保します。たとえば、本番データベースと開発データベースが混在する環境では、開発ワークロードに最大の QoS を適用し、本番ワークロードに最小の QoS を適用して、予測可能なパフォーマンスを確保します。

アダプティブ QoS

アダプティブ QoS は、ストレージ オブジェクトのサイズに基づいて制限を調整します。データベースで使用されることはほとんどありませんが (サイズはパフォーマンスのニーズと相関しないため)、パフォーマンス要件が容量に応じて変化する仮想化ワークロードには役立ちます。

効率性

ONTAP のスペース効率機能は、 ASA r2 システムに対して完全にサポートされ、最適化されています。ほとんどの場合、すべての効率化機能を有効にした状態でデフォルトのままにしておくことが最善のアプローチです。

ASA r2 システムはオールフラッシュSAN プラットフォームであるため、使用可能な容量を最大化し、コストを削減するには、圧縮、コンパクト化、重複排除などの効率化テクノロジが重要です。

圧縮

圧縮は、データ内のパターンをエンコードすることによってスペース要件を削減します。SSD ベースのASA r2 システムでは、フラッシュによってパフォーマンスのためのオーバープロビジョニングの必要性がなくなるため、圧縮によって大幅な節約が実現します。ONTAPアダプティブ圧縮はデフォルトで有効になっており、Oracle データベースを含むエンタープライズ ワークロードで徹底的にテストされており、レイテンシがマイクロ秒単位で測定される環境でも測定可能なパフォーマンスへの影響はありません。場合によっては、圧縮されたデータが占めるキャッシュスペースが少なくなるため、パフォーマンスが向上します。

温度に敏感なストレージ効率 (TSSE) は、 ASA r2 システムには適用されません。ASA r2 システムでは、圧縮はホット (頻繁にアクセスされる) データまたはコールド (あまりアクセスされない) データに基づいて行われません。データがコールドになるのを待たずに圧縮が開始されます。

適応圧縮

アダプティブ圧縮では、リレーショナル データベースで一般的に使用されるブロック サイズに一致する、デフォルトで 8 KB のブロック サイズが使用されます。ブロック サイズを大きくすると (16 KB または 32 KB)、トランザクション ログやバックアップなどのシーケンシャル データの効率が向上しますが、上書き時のオーバーヘッドを回避するために、アクティブなデータベースでは慎重に使用する必要があります。

ログやバックアップなどの静止ファイルの場合、ブロック サイズを最大 32 KB まで増やすことができます。デフォルトを変更する前に、 NetApp のガイダンスを参照してください。

ストリーミング バックアップでは重複排除による 32 KB 圧縮を使用しないでください。重複排除の効率を維持するために 8KB 圧縮を使用します。

圧縮のアライメント

ランダム上書きの場合、圧縮アライメントが重要になります。正しい LUN タイプ、パーティション オフセット (8 KB の倍数)、およびデータベース ブロック サイズに合わせたファイルシステム ブロック サイズを確認します。バックアップやログなどの連続データでは、アライメントを考慮する必要はありません。

データコンパクション

圧縮は、複数の圧縮ブロックが同じ物理ブロックを共有できるようにすることで、圧縮を補完します。たとえば、8KB のブロックが 1KB に圧縮される場合、圧縮によって残りのスペースが無駄にならないことが保証されます。この機能はインラインであり、パフォーマンスの低下を引き起こしません。

重複排除

重複排除はデータセット全体の重複ブロックを削除します。Oracle データベースでは通常、一意のブロックヘッダーとトレーラーにより重複排除による節約は最小限に抑えられますが、 ONTAP重複排除では、ゼロ化されたブロックと繰り返しパターンからスペースを再利用することができます。

効率性とシンプロビジョニング

ASA r2 システムは、デフォルトでシン プロビジョニングを使用します。効率化機能はシンプロビジョニングを補完し、使用可能な容量を最大化します。

ASA r2 ストレージ システムでは、ストレージ ユニットは常にシン プロビジョニングされます。シックプロビジョニングはサポートされていません。

クイックアシストテクノロジー（**QAT**）

NetApp ASA r2 プラットフォームでは、Intel QuickAssist テクノロジー (QAT) により、QAT を使用しないソフトウェア ベースの Temperature-Sensitive Storage Efficiency (TSSE) とは大きく異なるハードウェア アクセラレーションによる効率性が実現されます。

ハードウェア アクセラレーション付き **QAT:**

- 圧縮および暗号化タスクを CPU コアからオフロードします。
- ホット データ (頻繁にアクセスされるデータ) とコールド データ (あまりアクセスされないデータ) の両方に対して即時のインライン効率を実現します。
- CPU オーバーヘッドを大幅に削減します。
- より高いスループットとより低いレイテンシを実現します。
- TLS や VPN 暗号化などのパフォーマンスが重視される操作のスケーラビリティが向上します。

QATなしの**TSSE:**

- 効率的な操作のために CPU 駆動のプロセスに依存します。
- 遅延後のコールド データにのみ効率を適用します。
- より多くの CPU リソースを消費します。
- QAT アクセラレーション システムと比較して、全体的なパフォーマンスが制限されます。

したがって、最新のASA r2 システムは、古い TSSE のみのプラットフォームよりも高速で、ハードウェア アクセラレーションによる効率とシステム使用率が向上します。

ASA r2 の効率化のベストプラクティス

- NetAppの推奨事項*：

ASA r2のデフォルト

ASA r2 システムで実行されているONTAPで作成されたストレージ ユニットは、圧縮、コンパクション、重複排除などのすべてのインライン効率機能がデフォルトで有効になってシン プロビジョニングされます。Oracle データベースは一般に重複排除による大きなメリットを得られず、圧縮できないデータが含まれ

る場合もありますが、これらのデフォルト設定はほぼすべてのワークロードに適しています。ONTAP は、コスト削減につながるかどうかに関係なく、あらゆる種類のデータと I/O パターンを効率的に処理するように設計されています。デフォルトは、理由が完全に理解されており、逸脱することで明らかな利点がある場合にのみ変更する必要があります。

一般的な推奨事項

- 暗号化されたデータまたはアプリで圧縮されたデータの圧縮を無効にする: ファイルがすでにアプリケーション レベルで圧縮されているか暗号化されている場合は、圧縮を無効にしてパフォーマンスを最適化し、他のストレージ ユニットでより効率的な操作を可能にします。
- 大きな圧縮ブロックと重複排除を組み合わせないでください。データベースのバックアップに 32 KB の圧縮と重複排除の両方を使用しないでください。ストリーミング バックアップの場合、重複排除の効率を維持するために 8 KB 圧縮を使用します。
- 効率節約の監視: ONTAPツール (System Manager、 Active IQ) を使用して実際のスペース節約を追跡し、必要に応じてポリシーを調整します。

シンプロビジョニング

ASA r2 上の Oracle データベースのシン プロビジョニングでは、物理的に利用可能なスペースよりも多くの論理スペースを構成する必要があるため、慎重な計画が必要です。シン プロビジョニングを正しく実装すると、コストを大幅に削減し、管理性を向上させることができます。

シン プロビジョニングはASA r2 に不可欠であり、 ONTAP効率化テクノロジと密接に関連しています。どちらも、システムの物理容量よりも多くの論理データを保存できるためです。ASA r2 システムは SAN 専用であり、シン プロビジョニングはストレージ可用性ゾーン (SAZ) 内のストレージ ユニットと LUN に適用されます。

ASA r2 ストレージ ユニットは、デフォルトでシン プロビジョニングされます。

スナップショットのほぼすべての使用にはシン プロビジョニングが関係します。たとえば、30 日間のスナップショットを含む一般的な 10 TiB のデータベースは、論理データが 310 TiB として表示されますが、スナップショットには変更されたブロックのみが保存されるため、消費される物理スペースは 12 TiB ～ 15 TiB のみです。

同様に、クローン作成はシンプロビジョニングの別の形式です。80 TiB データベースのクローン 40 個を含む開発環境は、完全に書き込まれると 3.2 PiB 必要になりますが、実際には変更のみが保存されるため、消費量ははるかに少なくなります。

スペース管理

データの変更率が予期せず増加する可能性があるため、アプリケーション環境でのシン プロビジョニングには注意が必要です。たとえば、データベース テーブルのインデックスが再作成された場合や、VMware ゲストに大規模なパッチが適用された場合、スナップショットによるスペースの消費量が急速に増加する可能性があります。バックアップを誤って配置すると、非常に短時間で大量のデータが書き込まれる可能性があります。最後に、LUN の空き領域が予期せず不足すると、一部のアプリケーションの回復が困難になる可能性があります。

ASA r2 では、これらのリスクは、ボリューム自動拡張やスナップショット自動削除などのONTAP機能ではなく、シン プロビジョニング、プロアクティブ監視、および **LUN** サイズ変更ポリシー によって軽減されます。管理者は次のことを行う必要があります。

- LUN でシンプロビジョニングを有効にする (`space-reserve disabled`) - これはASA r2のデフォルト設定です
- System Manager アラートまたは API ベースの自動化を使用して容量を監視する
- 成長に合わせて、スケジュールまたはスクリプトによる LUN サイズ変更を使用する
- システムマネージャ（GUI）を使用してスナップショットの予約とSnapshotの自動作成の削除を構成する

> ASA r2 は自動ボリューム拡張や CLI 駆動型スナップショット削除をサポートしていないため、スペースしきい値と自動化スクリプトを慎重に計画することが重要です。

ASA r2 は、 WAFLベースのボリューム オプションを抽象化する SAN 専用のアーキテクチャであるため、部分予約設定を使用しません。代わりに、スペース効率と上書き保護は LUN レベルで管理されます。たとえば、ストレージ ユニットから 250 GiB の LUN がプロビジョニングされている場合、スナップショットは事前に同量のスペースを予約するのではなく、実際のブロックの変更に基づいてスペースを消費します。これにより、フラクショナル リザーブを使用する従来のONTAP環境で一般的だった大規模な静的予約が不要になります。

> 上書き保護の保証が必要であり、監視が実行できない場合は、管理者はストレージ ユニットに十分な容量をプロビジョニングし、スナップショットの予約を適切に設定する必要があります。ただし、 ASA r2 の設計では、ほとんどのワークロードで部分予約は不要になります。

圧縮と重複排除

ASA r2 の圧縮と重複排除は、従来のシン プロビジョニング メカニズムではなく、スペース効率化テクノロジです。これらの機能は、冗長データを排除し、ブロックを圧縮することで物理的なストレージフットプリントを削減し、本来の容量よりも多くの論理データを保存できるようにします。

たとえば、50 TiB のデータセットは 30 TiB に圧縮され、20 TiB の物理スペースを節約できます。アプリケーションの観点から見ると、ディスク上では 30 TiB しか占有していないにもかかわらず、まだ 50 TiB のデータが存在します。

> データセットの圧縮率は時間の経過とともに変化する可能性があり、物理的なスペースの消費量が増加する可能性があります。したがって、圧縮と重複排除は、監視と容量計画を通じて積極的に管理する必要があります。

空きスペースと**LVM**スペースの割り当て

削除されたブロックが再利用されない場合、 ASA r2 環境でのシン プロビジョニングは時間の経過とともに効率が低下する可能性があります。TRIM/UNMAP を使用してスペースを解放するか、ゼロで上書きしない限り (ASMRU - Automatic Space Management and Reclamation Utility 経由)、削除されたデータは物理容量を消費し続けます。多くの Oracle データベース環境では、データファイルは通常、作成時にフルサイズで事前割り当てされるため、シンプロビジョニングの利点は限られています。

LVM 構成を慎重に計画することで、効率が向上し、ストレージ プロビジョニングと LUN のサイズ変更の必要性が最小限に抑えられます。Veritas VxVM や Oracle ASM などの LVM を使用すると、基盤となる LUN は必要な場合にのみ使用されるエクステントに分割されます。たとえば、データセットのサイズが 2 TiB から始まり、時間の経過とともに 10 TiB に拡大する可能性がある場合には、このデータセットを LVM ディスク グループに編成された 10 TiB のシンプロビジョニングLUN に配置できます。作成時には 2 TiB のスペースのみを占有し、データの増加に対応するためにエクステントが割り当てられるときにのみ追加のスペースが必要になります。スペースが監視されている限り、このプロセスは安全です。

ONTAPフェイルオーバー

これらの操作中に Oracle データベースの操作が中断されないようにするには、ストレージ テイクオーバー機能について理解する必要があります。さらに、テイクオーバー操作で使用される引数は、誤って使用されるとデータの整合性に影響を与える可能性があります。

通常の状況では、特定のコントローラへの着信書き込みは、その HA パートナーに同期的にミラーリングされます。SnapMirror Active Sync (SM-as) を備えたASA r2 環境では、書き込みはセカンダリ サイトのリモート コントローラにもミラーリングされます。書き込みが不揮発性メディアのすべての場所に保存されるまで、ホスト アプリケーションに確認応答されません。

書き込みデータを保存するメディアは不揮発性メモリ (NVMEM) と呼ばれます。これは、不揮発性ランダム アクセス メモリ (NVRAM) と呼ばれることもあり、キャッシュではなく書き込みジャーナルと考えることができます。通常の動作中、 NVMEMからのデータは読み取られず、ソフトウェアまたはハードウェア障害が発生した場合にデータを保護するためだけに使用されます。データがドライブに書き込まれる場合、データはNVMEMからではなくシステム RAM から転送されます。

テイクオーバー操作中、HA ペアの 1 つのノードがパートナーから操作を引き継ぎます。ASA r2 では、MetroClusterがサポートされていないため、スイッチオーバーは適用されません。代わりに、 SnapMirror Active Sync によってサイト レベルの冗長性が提供されます。定期メンテナンス中のストレージ引き継ぎ操作は、ネットワーク パスの変更による操作の短時間の一時停止を除き、透過的である必要があります。ネットワークは複雑でエラーが発生しやすいため、 NetApp、ストレージ システムを実稼働させる前にテイクオーバー操作を徹底的にテストすることを強くお勧めします。そうすることが、すべてのネットワーク パスが正しく構成されていることを確認する唯一の方法です。SAN環境では、次のコマンドを使用してパスのステータスを確認します。 `sanlun lun show -p` または、オペレーティング システムのネイティブ マルチパス ツールを使用して、予想されるすべてのパスが利用可能であることを確認します。ASA r2 システムは LUN のすべてのアクティブな最適化されたパスを提供しますが、NVMe パスは sanlun によってカバーされていないため、NVMe 名前空間を使用する顧客は OS ネイティブ ツールに依存する必要があります。

強制買収を発行する際には注意が必要です。ストレージ構成を強制的に変更すると、ドライブを所有するコントローラの状態は無視され、代替ノードが強制的にドライブを制御することになります。強制テイクオーバーによってNVMEMの内容が破棄される可能性があるため、テイクオーバーを誤って強制すると、データの損失や破損が発生する可能性があります。引き継ぎが完了した後、そのデータが失われると、ドライブに保存されたデータは、データベースの観点から見ると少し古い状態に戻る可能性があります。

通常の HA ペアでは強制テイクオーバーが必要になることはほとんどありません。ほとんどすべての障害シナリオでは、ノードがシャットダウンし、自動フェイルオーバーが行われるようにパートナーに通知します。ノード間の相互接続が失われ、その後 1 つのコントローラーに障害が発生するローリング障害など、強制テイクオーバーが必要となるエッジ ケースがいくつかあります。このような状況では、コントローラの障害発生前にノード間のミラーリングが失われ、残りのコントローラには進行中の書き込みのコピーが残っていないことになります。その場合、強制的に引き継ぎを行う必要があり、データが失われる可能性があります。

NetApp、次の予防措置を講じることを推奨しています。

- 誤って強制的に買収を行わないよう十分注意してください。通常、強制する必要はありませんが、強制的に変更するとデータが失われる可能性があります。
- 強制テイクオーバーが必要な場合は、アプリケーションがシャットダウンされ、すべてのファイル システムがマウント解除され、論理ボリューム マネージャ(LVM) ボリューム グループが varyoff されていることを確認します。ASM ディスクグループをアンマウントする必要があります。
- SM-as の使用時にサイトレベルの障害が発生した場合、 ONTAP Mediator による自動計画外フェイルオーバーが存続クラスタで開始され、I/O が短時間停止した後、存続クラスタからデータベースの移行が続行されます。詳細については、 "ASA r2 システムでのSnapMirrorアクティブ同期" 詳細な設定手順については、こちらをご覧ください。

AFF/ FASシステムによるデータベース構成

ブロックサイズ

ONTAPでは内部的に可変ブロックサイズが使用されるため、Oracleデータベースには任意のブロックサイズを設定できます。ただし、ファイルシステムのブロックサイズがパフォーマンスに影響することがあり、場合によってはRedoブロックサイズを大きくすることでパフォーマンスが向上することがあります。

データファイルのブロックサイズ

OSによっては、ファイルシステムのブロックサイズを選択できます。Oracleデータファイルをサポートするファイルシステムでは、圧縮を使用する場合にブロックサイズを8KBにする必要があります。圧縮が不要な場合は、8KBまたは4KBのブロックサイズを使用できます。

512バイトのブロックを使用するファイルシステムにデータファイルが配置されていると、ファイルのミスアライメントが発生する可能性があります。LUNとファイルシステムはNetAppの推奨事項に基づいて適切にアライメントされていても、ファイルI/Oはミスアライメントされます。このようなミスアライメントが発生すると、原因で重大なパフォーマンス問題が発生します。

Redoログをサポートするファイルシステムでは、Redoブロックサイズの倍数のブロックサイズを使用する必要があります。そのためには、通常、RedoログファイルシステムとRedoログ自体の両方で512バイトのブロックサイズを使用する必要があります。

Redoブロックサイズ

Redo率が非常に高い場合は、少ない処理で効率よくI/Oを実行できるため、4KBのブロックサイズの方がパフォーマンスが向上する可能性があります。Redo率が50MBpsを超える場合は、4KBのブロックサイズをテストすることを検討してください。

一部のお客様では、ブロックサイズが512バイトのRedoログをブロックサイズが4KBのファイルシステムで使用し、非常に小さなトランザクションが大量に発生するという問題が報告されています。このパフォーマンスの問題は、複数の512バイトの変更を4KBの単一のファイルシステムブロックに適用する際のオーバーヘッドが原因で発生していましたが、ファイルシステムのブロックサイズを512バイトに変更することで解決されました。

* NetAppでは、関連するカスタマーサポートまたはプロフェッショナルサービス部門から指示があった場合、または正式な製品ドキュメントに基づく場合を除き、Redoブロックサイズを変更しないことを推奨しています*。

db_file_multiblock_read_count

。 `db_file_multiblock_read_count` パラメータは、シーケンシャルI/OでOracleが単一の処理として読み取るOracleデータベースブロックの最大数を制御します。

ただし、このパラメータは、すべての読み取り処理でOracleが読み取るブロック数には影響しません。また、ランダムI/Oにも影響しません。影響を受けるのはシーケンシャルI/Oのブロックサイズだけです。

Oracleでは、このパラメータを未設定のままにしておくことを推奨しています。これにより、データベースソフトウェアは自動的に最適な値を設定できます。つまり、このパラメータは通常、I/Oサイズが1MBになる値に設定されます。たとえば、8KBブロックの1MB読み取りでは128ブロックを読み取る必要があるため、このパラメータのデフォルト値は128になります。

顧客サイトのNetAppで発生したデータベースパフォーマンスの問題のほとんどには、このパラメータの設定が誤っています。Oracleバージョン8および9では、この値を変更する正当な理由があります。そのため、パラメータが `init.ora` ファイル：データベースがOracle 10以降にアップグレードされたため。従来の設定である8または16をデフォルト値の128と比較すると、シーケンシャルI/Oのパフォーマンスが大幅に低下します。

* NetApp推奨*設定 `db_file_multiblock_read_count` パラメータがに存在していてはなりません。 `init.ora` ファイル。NetAppでは、このパラメータを変更することでパフォーマンスが向上するという状況は発生していませんが、シーケンシャルI/Oのスループットに明らかな影響を及ぼすケースは少なくありません。

ファイルシステムオプション

Oracle初期化パラメータ `filesystemio_options` 非同期I/Oとダイレクト I/Oの使用を制御します。

一般的な考え方に反して、非同期I/Oとダイレクト I/Oは相互に排他的ではありません。NetAppでは、お客様の環境でこのパラメータの設定ミスが頻繁に発生し、この設定ミスが多くのパフォーマンス問題の直接的な原因となっていることを確認しています。

非同期I/Oとは、Oracle I/O処理を並行処理できることを意味します。さまざまなOSで非同期I/Oが使用可能になる前は、ユーザが多数のdbwriterプロセスを設定し、サーバプロセスの設定を変更していました。非同期I/Oでは、OS自体がデータベースソフトウェアに代わって効率的かつ並列的にI/Oを実行します。このプロセスによってデータがリスクにさらされることはなく、OracleのRedoロギングなどの重要な処理も同期的に実行されます。

ダイレクトI/OはOSのバッファキャッシュをバイパスします。UNIXシステムのI/Oは、通常、OSのバッファキャッシュを通過します。これは内部キャッシュを持たないアプリケーションでは便利ですが、OracleはSGA内に独自のバッファキャッシュを備えています。ほとんどの場合、OSのバッファキャッシュを使用するよりも、ダイレクトI/Oを有効にしてサーバRAMをSGAに割り当てる方が適しています。Oracle SGAはメモリをより効率的に使用します。さらに、I/OがOSバッファを通過すると、追加の処理が発生し、レイテンシが増加します。レイテンシの増加は、低レイテンシが重要な要件である書き込みI/Oの負荷が高い場合に特に顕著です。

オフシヨン `filesystemio_options` 次のとおりです。

- * async。* OracleはI/O要求をOSに送信して処理します。このプロセスにより、OracleはI/Oの完了を待たずに他の処理を実行できるため、I/Oの並列化が促進されます。
- * directio.* Oracleは、ホストOSキャッシュを介してI/Oをルーティングするのではなく、物理ファイルに対して直接I/Oを実行します。
- なし。 Oracleは同期I/Oとバッファ I/Oを使用します。この構成では、共有サーバプロセスと専用サーバプロセスの選択、およびdbwriterの数がより重要になります。
- * SETALL。* Oracleは非同期I/Oとダイレクト I/Oの両方を使用します。ほとんどすべての場合、 `setall` 最適です。

。 `filesystemio_options` パラメータは、DNFS環境とASM環境では効果がありません。DNFSまたはASMを使用すると、自動的に非同期I/Oとダイレクト I/Oの両方が使用されます。

一部のお客様では、特に以前のRed Hat Enterprise Linux 4（RHEL4）リリースで、過去に非同期I/Oの問題が発生していました。インターネット上のいくつかの時代遅れのアドバイスは、時代遅れの情報のために非同期IOを避けることを提案しています。非同期I/Oは、現在のすべてのOSで安定しています。OSの既知のバグがない限り、無効にする理由はありません。

データベースでバッファ I/Oが使用されている場合は、ダイレクト I/Oに切り替えてもSGAサイズの変更が必要になることがあります。バッファ I/Oを無効にすると、ホストOSキャッシュがデータベースに提供するパフォーマンス上のメリットがなくなります。RAMをSGAに再度追加すると、この問題が解決します。最終的には、I/Oパフォーマンスの向上につながります。

RAMはOSのバッファキャッシュよりもOracle SGAに使用する方がほとんどですが、最適な値を特定できない場合もあります。たとえば、断続的にアクティブになるOracleインスタンスが多数あるデータベースサーバでは、SGAサイズが非常に小さいバッファ I/Oを使用することを推奨します。この方法では、実行中のすべてのデータベースインスタンスが、空いているOSのRAMを柔軟に使用できます。これは非常にまれな状況ですが、一部のお客様のサイトで確認されています。

* NetApp推奨*設定 `filesystemio_options` 終了： `setall`ただし、状況によっては、ホストのバッファキャッシュが失われた場合にOracle SGAの拡張が必要になることがあります。

RACタイムアウト

Oracle RACは、クラスタの健常性を監視する複数のタイプの内部ハートビートプロセスを備えたクラスタウェア製品です。

の情報 "MissCount" セクションには、ネットワーク・ストレージを使用するOracle RAC環境に関する重要な情報が記載されています。多くの場合、RACクラスタがネットワーク・パスの変更やストレージのフェイルオーバー/スイッチオーバー操作に耐えられるように、デフォルトのOracle RAC設定を変更する必要があります。

ディスクタイムアウト

プライマリストレージ関連のRACパラメータはです。 `disktimeout`。このパラメータは、投票ファイルI/Oが完了しなければならないしきい値を制御します。状況に応じて `disktimeout` パラメータの値を超えると、そのRACノードがクラスタから削除されます。このパラメータのデフォルトは200です。ストレージのテイクオーバーとギブバックの標準的な手順では、この値で十分です。

テイクオーバーやギブバックには多くの要素が影響するため、NetAppでは、RAC構成を本番環境に導入する前に徹底的にテストすることを強く推奨します。ストレージフェイルオーバーの完了に必要な時間に加えて、Link Aggregation Control Protocol（LACP；リンクアグリゲーション制御プロトコル）の変更が伝播されるまでの時間も長くなります。また、SANマルチパスソフトウェアはI/Oタイムアウトを検出し、代替パスで再試行する必要があります。データベースが非常にアクティブな場合は、投票ディスクI/Oが処理される前に、大量のI/Oをキューに入れて再試行する必要があります。

ストレージのテイクオーバーやギブバックを実際に実行できない場合は、データベースサーバでケーブルを取り外すテストを実行して影響をシミュレートできます。

- NetAppの推奨事項*：
- を終了します。 `disktimeout` パラメータを指定します。デフォルト値は200です。
- RAC構成は常に十分にテストしてください。

MissCount

。 `misscount` パラメータは通常、RACノード間のネットワークハートビートにのみ影響します。デフォルトは30秒です。Gridバイナリがストレージアレイ上にある場合やOSのブートドライブがローカルでない場合は、このパラメータが重要になることがあります。これには、ブートドライブがFC SANに配置されたホスト、NFSブートOS、およびVMDKファイルなどの仮想データストアに配置されたブートドライブが含まれます。

ブートドライブへのアクセスがストレージのテイクオーバーやギブバックによって中断された場合、Gridバイナリの場所またはOS全体が一時的に停止する可能性があります。ONTAPがストレージ処理を完了するのに必要な時間、およびOSがパスを変更してI/Oを再開するのに必要な時間が、 `misscount` しきい値。そのため、ブートLUNまたはGridバイナリへの接続がリストアされたあと、ノードはただちに削除されます。ほとんどの場合、削除とその後のリブートは実行されますが、リブートの理由を示すログメッセージは表示されません。すべての構成に影響するわけではないので、RAC環境内のSANブート、NFSブート、またはデータストアベースのホストをテストして、ブートドライブへの通信が中断してもRACが安定した状態になるようにします。

ローカルでないブートドライブまたはローカルでないファイルシステムをホストしている場合 `grid` バイナリ、 `misscount` 一致するように変更する必要があります `disktimeout`。このパラメータを変更した場合は、さらにテストを行い、ノードのフェイルオーバー時間など、RACの動作への影響を特定します。

- NetAppの推奨事項*：
- そのままにします。 `misscount` 次のいずれかの条件が適用されない場合は、デフォルト値の30のパラメータを使用します。
  - `grid` バイナリが、ネットワークに接続されたドライブ（NFS、iSCSI、FC、データストアベースのドライブを含む）に配置されている。
  - OSがSANブートである。
- このような場合は、ネットワークの中断がOSへのアクセスに影響するか、 `GRID_HOME` ファイルシステム：このような中断によって原因Oracle RACデーモンが停止し、 `misscount`-ベースのタイムアウトおよび削除。タイムアウトのデフォルトは27秒です。これは `misscount` マイナス `reboottime`。このような場合、 `misscount` 200にして一致させる `disktimeout`。

ASA r2 システムによるデータベース構成

ブロックサイズ

ONTAP は内部的に可変ブロック サイズを使用するため、Oracle データベースは任意のブロック サイズで構成できます。ただし、ファイルシステムのブロック サイズはパフォーマンスに影響を与える可能性があり、場合によっては、REDO ブロック サイズを大きくするとパフォーマンスが向上することがあります。

ASA r2 では、 AFF/ FASシステムと比較して、Oracle ブロック サイズの推奨事項に変更はありません。ONTAP の動作はすべてのプラットフォームで一貫しています。

データファイルのブロックサイズ

OSによっては、ファイルシステムのブロックサイズを選択できます。Oracleデータファイルをサポートするファイルシステムでは、圧縮を使用する場合にブロックサイズを8KBにする必要があります。圧縮が不要な場合は、8KBまたは4KBのブロックサイズを使用できます。

512バイトのブロックを使用するファイルシステムにデータファイルが配置されていると、ファイルのミスアライメントが発生する可能性があります。LUNとファイルシステムはNetAppの推奨事項に基づいて適切にアライメントされていても、ファイルI/Oはミスアライメントされます。このようなミスアライメントが発生すると、原因で重大なパフォーマンス問題が発生します。

Redoブロックサイズ

Redoログをサポートするファイルシステムでは、Redoブロックサイズの倍数のブロックサイズを使用する必要があります。そのためには、通常、RedoログファイルシステムとRedoログ自体の両方で512バイトのブロックサイズを使用する必要があります。

Redo率が非常に高い場合は、少ない処理で効率よくI/Oを実行できるため、4KBのブロックサイズの方がパフォーマンスが向上する可能性があります。Redo率が50MBpsを超える場合は、4KBのブロックサイズをテストすることを検討してください。

一部のお客様では、ブロックサイズが512バイトのRedoログをブロックサイズが4KBのファイルシステムで使用し、非常に小さなトランザクションが大量に発生するという問題が報告されています。このパフォーマンスの問題は、複数の512バイトの変更を4KBの単一のファイルシステムブロックに適用する際のオーバーヘッドが原因で発生していましたが、ファイルシステムのブロックサイズを512バイトに変更することで解決されました。

* NetAppでは、関連するカスタマーサポートまたはプロフェッショナルサービス部門から指示があった場合、または正式な製品ドキュメントに基づく場合を除き、Redoブロックサイズを変更しないことを推奨しています*。

db_file_multiblock_read_count

。 `db_file_multiblock_read_count` パラメータは、シーケンシャルI/OでOracleが単一の処理として読み取るOracleデータベースブロックの最大数を制御します。

AFF/ FASシステムと比較して推奨事項に変更はありません。ONTAP の動作と Oracle のベスト プラクティスは、 ASA r2、 AFF、およびFASプラットフォーム全体で同一です。

ただし、このパラメータは、すべての読み取り処理でOracleが読み取るブロック数には影響しません。また、ランダムI/Oにも影響しません。影響を受けるのはシーケンシャルI/Oのブロックサイズだけです。

Oracleでは、このパラメータを未設定のままにしておくことを推奨しています。これにより、データベースソフトウェアは自動的に最適な値を設定できます。つまり、このパラメータは通常、I/Oサイズが1MBになる値に設定されます。たとえば、8KBブロックの1MB読み取りでは128ブロックを読み取る必要があるため、このパラメータのデフォルト値は128になります。

顧客サイトのNetAppで発生したデータベースパフォーマンスの問題のほとんどには、このパラメータの設定が誤っています。Oracleバージョン8および9では、この値を変更する正当な理由があります。そのため、パラメータが `init.ora` ファイル：データベースがOracle 10以降にアップグレードされたため。従来の設定である8または16をデフォルト値の128と比較すると、シーケンシャルI/Oのパフォーマンスが大幅に低下します。

* NetApp推奨*設定 `db_file_multiblock_read_count` パラメータがに存在していてはなりません。 `init.ora` ファイル。NetAppでは、このパラメータを変更することでパフォーマンスが向上するという状況は発生していませんが、シーケンシャルI/Oのスループットに明らかな影響を及ぼすケースは少なくありません。

ファイルシステムオプション

Oracle初期化パラメータ `filesystemio_options` 非同期I/OとダイレクトI/Oの使用を制御します。

ASA r2 の filesystemio_options の動作と推奨事項は、パラメータが Oracle 固有であり、ストレージ プラットフォームに依存しないため、 AFF/ FASシステムと同じです。ASA r2 はAFF/ FASと同様にONTAP を使用するため、同じベスト プラクティスが適用されます。

一般的な考え方に反して、非同期I/OとダイレクトI/Oは相互に排他的ではありません。NetAppでは、お客様の環境でこのパラメータの設定ミスが頻繁に発生し、この設定ミスが多くのパフォーマンス問題の直接的な原因となっていることを確認しています。

非同期I/Oとは、Oracle I/O処理を並行処理できることを意味します。さまざまなOSで非同期I/Oが使用可能になる前は、ユーザが多数のdbwriterプロセスを設定し、サーバプロセスの設定を変更していました。非同期I/Oでは、OS自体がデータベースソフトウェアに代わって効率的かつ並列的にI/Oを実行します。このプロセスによってデータがリスクにさらされることはなく、OracleのRedoロギングなどの重要な処理も同期的に実行されます。

ダイレクトI/OはOSのバッファキャッシュをバイパスします。UNIXシステムのI/Oは、通常、OSのバッファキャッシュを通過します。これは内部キャッシュを持たないアプリケーションでは便利ですが、OracleはSGA内に独自のバッファキャッシュを備えています。ほとんどの場合、OSのバッファキャッシュを使用するよりも、ダイレクトI/Oを有効にしてサーバRAMをSGAに割り当てる方が適しています。Oracle SGAはメモリをより効率的に使用します。さらに、I/OがOSバッファを通過すると、追加の処理が発生し、レイテンシが増加します。レイテンシの増加は、低レイテンシが重要な要件である書き込みI/Oの負荷が高い場合に特に顕著です。

オフシヨン `filesystemio_options` 次のとおりです。

- * async。* OracleはI/O要求をOSに送信して処理します。このプロセスにより、OracleはI/Oの完了を待たずに他の処理を実行できるため、I/Oの並列化が促進されます。
- * directio.* Oracleは、ホストOSキャッシュを介してI/Oをルーティングするのではなく、物理ファイルに対して直接I/Oを実行します。

- なし。 Oracleは同期I/Oとバッファ I/Oを使用します。この構成では、共有サーバプロセスと専用サーバプロセスの選択、およびdbwriterの数がより重要になります。
- * SETALL。* Oracleは非同期I/Oとダイレクト I/Oの両方を使用します。ほとんどすべての場合、 `setall` 最適です。

> ASM環境では、OracleはASM管理ディスクに対して直接I/Oと非同期I/Oを自動的に使用するため、 `filesystemio_options` ASM ディスク グループには影響しません。ASM 以外の展開 (SAN LUN 上のファイル システムなど) の場合は、次のように設定します。 `filesystemio_options = setall`。これにより、非同期 I/O と直接 I/O の両方が有効になり、最適なパフォーマンスが得られます。

一部の古いオペレーティング システムでは非同期I/Oに問題があったため、非同期 I/O を避けるように勧める古いアドバイスがありました。ただし、非同期I/Oは安定しており、現在のすべてのオペレーティング システムで完全にサポートされています。特定の OS バグが特定されない限り、無効にする理由はありません。

データベースでバッファI/Oが使用されている場合は、ダイレクトI/Oに切り替えてもSGAサイズの変更が必要になることがあります。バッファI/Oを無効にすると、ホストOSキャッシュがデータベースに提供するパフォーマンス上のメリットがなくなります。RAMをSGAに再度追加すると、この問題が解決します。最終的には、I/Oパフォーマンスの向上につながります。

RAMはOSのバッファキャッシュよりもOracle SGAに使用する方がほとんどですが、最適な値を特定できない場合もあります。たとえば、断続的にアクティブになるOracleインスタンスが多数あるデータベースサーバでは、SGAサイズが非常に小さいバッファI/Oを使用することを推奨します。この方法では、実行中のすべてのデータベースインスタンスが、空いているOSのRAMを柔軟に使用できます。これは非常にまれな状況ですが、一部のお客様のサイトで確認されています。

> * NetAppの推奨設定* `filesystemio_options` に `setall`ただし、状況によっては、ホスト バッファ キャッシュの損失により Oracle SGA の増加が必要になる場合があることに注意してください。ASA r2 システムは、低レイテンシの SAN ワークロード向けに最適化されているため、setall の使用は、ハイパフォーマンスのOracle 展開向けの ASA の設計と完全に一致します。

RACタイムアウト

Oracle RACは、クラスタの健常性を監視する複数のタイプの内部ハートビートプロセスを備えたクラスタウェア製品です。

ASA r2 システムはAFF/ FASと同様にONTAP を使用するため、Oracle RAC タイムアウト パラメータにも同じ原則が適用されます。disktimeout または misscount の推奨事項にはASA固有の変更はありません。ただし、 ASA r2 は SAN ワークロードと低レイテンシフェイルオーバー向けに最適化されているため、これらのベスト プラクティスはさらに重要になります。

> の情報は "MissCount" このセクションには、ネットワーク ストレージを使用する Oracle RAC 環境に関する重要な情報が含まれています。多くの場合、RAC クラスタがネットワーク パスの変更やストレージ フェイルオーバー操作に耐えられるように、デフォルトの Oracle RAC 設定を変更する必要があります。

ディスクタイムアウト

プライマリストレージ関連のRACパラメータはです。 `disktimeout`。このパラメータは、投票ファイルI/O

が完了しなければならないしきい値を制御します。状況に応じて `disktimeout` パラメータの値を超えると、そのRACノードがクラスタから削除されます。このパラメータのデフォルトは200です。ストレージのテイクオーバーとギブバックの標準的な手順では、この値で十分です。

テイクオーバーやギブバックには多くの要素が影響するため、NetAppでは、RAC構成を本番環境に導入する前に徹底的にテストすることを強く推奨します。ストレージフェイルオーバーの完了に必要な時間に加えて、Link Aggregation Control Protocol（LACP；リンクアグリゲーション制御プロトコル）の変更が伝播されるまでの時間も長くなります。また、SANマルチパスソフトウェアはI/Oタイムアウトを検出し、代替パスで再試行する必要があります。データベースが非常にアクティブな場合は、投票ディスクI/Oが処理される前に、大量のI/Oをキューに入れて再試行する必要があります。

ストレージのテイクオーバーやギブバックを実際に実行できない場合は、データベースサーバでケーブルを取り外すテストを実行して影響をシミュレートできます。

- NetAppの推奨事項*：
- を終了します。 `disktimeout` パラメータを指定します。デフォルト値は200です。
- RAC構成は常に十分にテストしてください。

MissCount

。 `misscount` パラメータは通常、RACノード間のネットワークハートビートにのみ影響します。デフォルトは30秒です。Gridバイナリがストレージアレイ上にある場合やOSのブートドライブがローカルでない場合は、このパラメータが重要になることがあります。これには、ブートドライブがFC SANに配置されたホスト、NFSブートOS、およびVMDKファイルなどの仮想データストアに配置されたブートドライブが含まれます。

ブートドライブへのアクセスがストレージのテイクオーバーやギブバックによって中断された場合、Gridバイナリの場所またはOS全体が一時的に停止する可能性があります。ONTAPがストレージ処理を完了するのに必要な時間、およびOSがパスを変更してI/Oを再開するのに必要な時間が、 `misscount` しきい値。そのため、ブートLUNまたはGridバイナリへの接続がリストアされたあと、ノードはただちに削除されます。ほとんどの場合、削除とその後のリブートは実行されますが、リブートの理由を示すログメッセージは表示されません。すべての構成に影響するわけではないので、RAC環境内のSANブート、NFSブート、またはデータストアベースのホストをテストして、ブートドライブへの通信が中断してもRACが安定した状態になるようにします。

ローカルでないブートドライブまたはローカルでないファイルシステムをホストしている場合 `grid` バイナリ、 `misscount` 一致するように変更する必要があります `disktimeout`。このパラメータを変更した場合は、さらにテストを行い、ノードのフェイルオーバー時間など、RACの動作への影響を特定します。

- NetAppの推奨事項*：
- そのままにします。 `misscount` 次のいずれかの条件が適用されない場合は、デフォルト値の30のパラメータを使用します。
  - `grid` バイナリは、iSCSI、FC、データストアベースのドライブなどのネットワーク接続ドライブ上に配置されます。
  - OSがSANブートである。
- このような場合は、ネットワークの中断がOSへのアクセスに影響するか、 `GRID_HOME` ファイルシステム：このような中断によって原因Oracle RACデーモンが停止し、 `misscount`-ベースのタイムアウトおよび削除。タイムアウトのデフォルトは27秒です。これは `misscount` マイナス `reboottime`。このような場合、 `misscount` 200にして一致させる `disktimeout`。

- ASA r2 の SAN 最適化設計によりフェイルオーバーの遅延は短縮されますが、ネットワーク ブートまたはグリッド バイナリのタイムアウトは依然として調整する必要があります。
- 拡張 RAC またはアクティブ / アクティブセットアップ (SnapMirrorアクティブ シンクなど) の場合、ゼロ RPO アーキテクチャではタイムアウトの調整が依然として重要です。

AFF/ FASシステムを使用したホスト構成

AIX の場合

ONTAPを使用したIBM AIX上のOracleデータベースの構成に関するトピック。

同時I/O

IBM AIXで最適なパフォーマンスを実現するには、同時I/Oを使用する必要があります。AIXはシリアル化されたアトミックなI/Oを実行するため、大量のオーバーヘッドが発生するため、同時I/Oがないとパフォーマンスが制限される可能性があります。

従来のNetAppでは、 `cio` マウントオプション：ファイルシステムで強制的に同時I/Oを使用しますが、このプロセスには欠点があるため不要になりました。AIX 5.2とOracle 10gR1が導入されて以降、AIX上のOracleでは、ファイルシステム全体で同時I/Oを強制的に実行するのではなく、個 々 のファイルを開いて同時IOを実行できるようになりました。

同時I/Oを有効にする最適な方法は、 `init.ora` パラメータ `filesystemio_options` 終了： `setall`。これにより、Oracleが特定のファイルを開いて同時I/Oで使用できるようになります。

を使用します `cio` マウントオプションを指定すると、同時I/Oが強制的に使用されるため、悪影響が生じる可能性があります。たとえば、同時I/Oを強制するとファイルシステムの先読みが無効になり、Oracleデータベースソフトウェアの外部で発生するI/O（ファイルのコピーやテープバックアップの実行など）のパフォーマンスが低下する可能性があります。さらに、Oracle GoldenGateやSAP BR * Toolsなどの製品は、 `cio` 特定のバージョンのOracleでのマウントオプション。

- NetAppの推奨事項*：
- を使用しないでください `cio` ファイルシステムレベルのマウントオプション。代わりに、を使用して同時I/Oを有効にします。 `filesystemio_options=setall`。
- 使用するのは、 `cio` マウントオプションは次のように設定できない場合に実行します： `filesystemio_options=setall`。

AIX NFSのマウントオプション

次の表に、OracleシングルインスタンスデータベースのAIX NFSマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
制御ファイル データファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, intr`

次の表に、RACのAIX NFSマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
制御ファイル データファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac`
`CRS/Voting`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac`
専用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
共有 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr`

シングルインスタンスとRACマウントオプションの主な違いは、 `noac` をマウントオプションに移動します。このオプションを使用するとホストOSのキャッシングが無効になるため、データの状態について、RACクラスタ内のすべてのインスタンスが一貫した情報を認識できるようになります。

ただし、 `cio` マウントオプションと `init.ora` パラメータ `filesystemio_options=setall` ホストのキャッシングを無効にした場合と同じ効果がありますが、引き続きを使用する必要があります。 `noac`。 `noac` 共有の場合は必須です `ORACLE_HOME` Oracleパスワードファイルやパスワードファイルなどのファイルの整合性を維持するための導入 `spfile` パラメータファイル。RACクラスタ内の各インスタンスに専用の

`ORACLE_HOME`の場合、このパラメータは必要ありません。

AIX JFS / JFS2のマウントオプション

次の表に、AIX JFS / JFS2のマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	デフォルト値です
制御ファイル データファイル REDO ログ	デフォルト値です
ORACLE_HOME を参照してください	デフォルト値です

AIXを使用する前に `hdisk` データベースを含むあらゆる環境のデバイスで、パラメータをチェックします。`queue_depth`。このパラメータはHBAのキュー深度ではなく、個 々 ののSCSIキュー深度に関連します。`hdisk device. Depending on how the LUNs are configured, the value for `queue_depth` パフォーマンスを向上させるには低すぎる可能性があります。テストでは、最適値は64であることが示されています。

HP-UX

ONTAPを使用したHP-UX上のOracleデータベースの設定に関するトピック。

HP-UX NFSのマウントオプション

次の表に、単一インスタンスのHP-UX NFSマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid`
制御ファイル データファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,forcedirectio, nointr,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid`

次の表に、RACのHP-UX NFSマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,noac,suid`

ファイルタイプ	マウントオプション
制御ファイル データファイル REDO ログ	`rw, bg,hard, [vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio,suid`
CRS /投票	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac, forcedirectio,suid`
専用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
共有 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,suid`

シングルインスタンスとRACマウントオプションの主な違いは、 `noac` および `forcedirectio` をマウントオプションに移動します。このオプションを使用するとホストOSのキャッシングが無効になるため、データの状態について、RACクラスタ内のすべてのインスタンスが一貫した情報を認識できるようになります。ただし、 `init.ora` パラメータ `filesystemio_options=setall` ホストのキャッシングを無効にした場合と同じ効果がありますが、引き続きを使用する必要があります。 `noac` および `forcedirectio`。

理由 `noac` 共有の場合は必須です `ORACLE_HOME` を導入すると、Oracleパスワードファイルやspfileなどのファイルの整合性が維持されます。RACクラスタ内の各インスタンスに専用の `ORACLE_HOME`、このパラメータは必須ではありません。

HP-UX VxFSマウントオプション

Oracleバイナリをホストするファイルシステムには、次のマウントオプションを使用します。

```
delaylog,nodatainlog
```

データファイル、Redoログ、アーカイブログ、制御ファイルが格納されているファイルシステムで、HP-UXのバージョンが同時I/Oをサポートしていない場合は、次のマウントオプションを使用します。

```
nodatainlog,mincache=direct,convosync=direct
```

同時I/Oがサポートされている場合（VxFS 5.0.1以降、またはServiceGuard Storage Management Suiteを使用している場合）は、データファイル、Redoログ、アーカイブログ、および制御ファイルを格納しているファイルシステムで次のマウントオプションを使用します。

```
delaylog,cio
```

パラメータ `db_file_multiblock_read_count` VxFS環境では特に重要です。Oracleでは、特に指示がないかぎり、Oracle 10g R1以降ではこのパラメータを未設定のままにすることを推奨しています。Oracleの8KBブロックサイズの場合、デフォルトは128です。このパラメータの値が強制的に16以下になる場合は、 `convosync=direct` シーケンシャルI/Oのパフォーマンスが低下する可能性があるため、マウントオプションを使用します。この手順は、パフォーマンスの他の側面に損傷を与えるため、 `db_file_multiblock_read_count` デフォルト値から変更する必要があります。

Linux の場合

Linux OSに固有の設定に関するトピック。

Linux NFSv3 TCPスロットテーブル

TCPスロットテーブルは、NFSv3でホストバスアダプタ（HBA）のキュー深度に相当します。一度に未処理となることのできるNFS処理の数を制御します。デフォルト値は通常16ですが、最適なパフォーマンスを得るには小さすぎます。逆に、新しいLinuxカーネルでTCPスロットテーブルの上限をNFSサーバが要求でいっぱいになるレベルに自動的に引き上げることができるため、問題が発生します。

パフォーマンスを最適化し、パフォーマンスの問題を回避するには、TCPスロットテーブルを制御するカーネルパラメータを調整します。

を実行します `sysctl -a | grep tcp.*.slot_table` コマンドを実行し、次のパラメータを確認します。

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

すべてのLinuxシステムに `sunrpc.tcp_slot_table_entries`ただし、次のようなものがあります。 `sunrpc.tcp_max_slot_table_entries`。どちらも128に設定する必要があります。

これらのパラメータを設定しないと、パフォーマンスに大きく影響する可能性があります。Linux OSが十分なI/Oを発行していないためにパフォーマンスが制限される場合もあります。一方では、Linux OSが問題で処理できる以上のI/Oを試行すると、I/Oレイテンシが増加します。

Linux NFSのマウントオプション

次の表に、単一インスタンスのLinux NFSのマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
制御ファイル データファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr`

ファイルタイプ	マウントオプション
ORACLE_HOME を参照してください	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr`

次の表に、RACのLinux NFSマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,actimeo=0`
制御ファイル データ・ファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,actimeo=0`
CRS /投票	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,actimeo=0`
専用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`
共有 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,actimeo=0`

シングルインスタンスとRACマウントオプションの主な違いは、 `actimeo=0` をマウントオプションに移動します。このオプションを使用するとホストOSのキャッシングが無効になるため、データの状態について、RACクラスタ内のすべてのインスタンスが一貫した情報を認識できるようになります。ただし、 `init.ora` パラメータ `filesystemio_options=setall` ホストのキャッシングを無効にした場合と同じ効果がありますが、引き続きを使用する必要があります。 `actimeo=0`。

理由 `actimeo=0` 共有の場合は必須です `ORACLE_HOME` を導入すると、Oracleパスワードファイルやspfileなどのファイルの整合性が維持されます。RACクラスタ内の各インスタンスに専用の `ORACLE_HOME`の場合、このパラメータは必要ありません。

一般に、データベース以外のファイルは、シングルインスタンスのデータファイルと同じオプションを使用してマウントします。ただしアプリケーションによっては要件が異なる場合があります。マウントオプションを使用しない `noac` および `actimeo=0` これらのオプションは、ファイルシステムレベルの先読みとバッファリングを無効にするため、可能であれば可能です。これにより、原因抽出、変換、ロードなどのプロセスで重大なパフォーマンスの問題が発生する可能性があります。

ACCESSとGETATTR

一部のお客様は、ACCESSやGETATTRなどのIOPSがワークロードを占有する可能性が非常に高いことを指摘しています。極端なケースでは、読み取りや書き込みなどの処理が全体の10%にまで低下することがあります。これは、を含むデータベースでは正常に動作します。 `actimeo=0` および / または `noac` Linuxの場合：これらのオプションは、Linux OSを原因して、ストレージシステムからファイルメタデータを定期的にリロードします。ACCESSやGETATTRなどの処理は影響力の低い処理で、データベース環境ではONTAPキャッシュから処理されます。読み取りや書き込みなど、ストレージシステムに真の需要を生み出す純粋なIOPSとみなすべきではありません。ただし、特にRAC環境では、これらのIOPSにはある程度の負荷がかかります。この状況に対処するには、DNFSを有効にして、OSのバッファキャッシュをバイパスし、不要なメタデータ処

理を回避します。

Linux Direct NFS

もう1つのマウントオプション（ nosharecache`は、（a）DNFSが有効で、（b）1つのソースボリュームが1つのサーバ（c）に複数回マウントされ、NFSマウントがネストされている場合に必要です。この構成は、主にSAPアプリケーションをサポートしている環境で見られます。たとえば、NetAppシステム上の1つのボリュームに、次の場所にディレクトリを配置できます。 `/vol/oracle/base 1秒前に /vol/oracle/home。状況 /vol/oracle/base はにマウントされます。 /oracle および /vol/oracle/home はにマウントされます。 `/oracle/home`を指定すると、同じソースからのNFSマウントがネストされます。

OSは、とが /oracle/home`同じボリューム（同じソースファイルシステム）に存在することを検出できます `/oracle。その後、OSは同じデバイスハンドルを使用してデータにアクセスします。これにより、OSキャッシングなどの特定の処理の使用が改善されますが、DNFSの妨げになります。DNFSがの /oracle/home`ようなファイルにアクセスする必要がある場合 `spfile、誤ってデータへの間違ったパスを使用しようとする可能性があります。その結果、I/O処理が失敗します。このような構成では、ソースボリュームをそのホスト上の別のNFSファイルシステムと共有するすべてのNFSファイルシステムに、マウントオプションを追加します nosharecache。これにより、Linux OSはそのファイルシステムに独立したデバイスハンドルを割り当てるようになります。

Linux Direct NFSとOracle RAC

Linux OS上のOracle RACでは、ノード間の一貫性を維持するためにRACで必要となるダイレクトI/Oを強制的に実行する方法がLinuxにないため、DNFSを使用するとパフォーマンスが特別に向上します。Linuxを回避策として使用するには、 actimeo=0 マウントオプション。OSキャッシュからファイルデータがただちに期限切れになります。このオプションを使用すると、Linux NFSクライアントは属性データを定期的に再読み取りするため、レイテンシが低下し、ストレージコントローラの負荷が増加します。

DNFSを有効にすると、ホストNFSクライアントがバイパスされ、この被害を回避できます。DNFSを有効にしたところ、RACクラスタのパフォーマンスが大幅に向上し、（特に他のIOPSに関して）ONTAPの負荷が大幅に減少したという報告が複数のお客様から寄せられています。

Linux Direct NFSとoranfstabファイル

マルチパスオプションを指定してLinuxでDNFSを使用する場合は、複数のサブネットを使用する必要があります。他のOSでは、を使用して複数のDNFSチャネルを確立できます。 LOCAL および DONTROUTE 単一のサブネット上に複数のDNFSチャネルを設定するためのオプション。ただし、これはLinuxでは正しく機能せず、予期しないパフォーマンスの問題が発生する可能性があります。Linuxでは、DNFSトラフィックに使用するNICをそれぞれ別 々 のサブネットに配置する必要があります。

I/Oスケジューラ

Linuxカーネルでは、ブロックデバイスへのI/Oのスケジュール方法を低レベルで制御できます。デフォルト値はLinuxのディストリビューションによって大きく異なります。テストでは、通常はDeadlineが最良の結果を提供することが示されていますが、場合によってはNOOPがわずかに改善されています。パフォーマンスの違いはごくわずかですが、データベース構成から最大限のパフォーマンスを引き出す必要がある場合は、両方のオプションをテストしてください。CFQは多くの構成でデフォルトであり、データベースワークロードのパフォーマンスに重大な問題があることが実証されています。

I/Oスケジューラの設定手順については、該当するLinuxベンダーのドキュメントを参照してください。

一部のお客様では、マルチパスデーモンがシステムで実行されていなかったために、ネットワーク停止中にクラッシュが発生しました。最近のバージョンのLinuxでは、OSとマルチパスデーモンのインストールプロセスによって、これらのOSがこの問題に対して脆弱なままになる可能性があります。パッケージは正しくインストールされていますが、再起動後の自動起動が設定されていません。

たとえば、RHEL5.5のマルチパスデーモンのデフォルトは次のようになります。

```
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

これを修正するには、次のコマンドを使用します。

```
[root@host1 iscsi]# chkconfig multipathd on
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ASMミラーリング

ASM ミラーリングでは、 ASM が問題を認識して代替の障害グループに切り替えるために、 Linux マルチパス設定の変更が必要になる場合があります。ONTAP 上のほとんどの ASM 構成では、外部冗長性が使用されます。つまり、データ保護は外部アレイによって提供され、 ASM はデータをミラーリングしません。一部のサイトでは、通常の冗長性を備えた ASM を使用して、通常は異なるサイト間で双方向ミラーリングを提供しています。

に表示されるLinux設定 "NetApp Host Utilitiesのマニュアル" I/Oが無期限にキューイングされるマルチパスパラメータを指定します。つまり、アクティブなパスがないLUNデバイス上のI/Oは、I/Oが完了するまで待機します。これは、SANパスの変更が完了するまで、FCスイッチがリブートするまで、またはストレージシステムがフェイルオーバーを完了するまで、Linuxホストが必要な時間だけ待機するために、通常は推奨されます。

この無制限のキューイング動作により、ASMミラーリングで問題が発生します。ASMは、代替LUNでI/Oを再試行するためにI/O障害を受信する必要があるためです。

Linuxで次のパラメータを設定します。 `multipath.conf` ASMミラーリングで使用されるASM LUNのファイル：

```
polling_interval 5
no_path_retry 24
```

これらの設定により、ASMデバイスに120秒のタイムアウトが作成されます。タイムアウトは、`polling_interval` * `no_path_retry` 秒として。状況によっては正確な値の調整が必要になる場合がありますが、ほとんどの場合は120秒のタイムアウトで十分です。具体的には、コントローラのテイクオーバーまたはギブバックが120秒以内に実行され、I/Oエラーが発生しないようにしてください。この場合、障害グループはオフラインになります。

A下限 `no_path_retry` この値を指定すると、ASMが代替障害グループに切り替えるのに必要な時間を短縮

できますが、これにより、コントローラのテイクオーバーなどのメンテナンス作業中に不要なフェイルオーバーが発生するリスクも高まります。ASMミラーリングの状態を注意深く監視することで、このリスクを軽減できます。不要なフェイルオーバーが発生した場合、再同期が比較的短時間で実行されると、ミラーを迅速に再同期できます。追加情報については、使用しているOracleソフトウェアのバージョンに対応するASM高速ミラー再同期に関するOracleのマニュアルを参照してください。

Linuxの**xfs**、**ext3**、**ext4**のマウントオプション

* NetAppでは*デフォルトのマウントオプションを使用することを推奨しています。

ASMlib/AFD（ASMフィルタドライバ）

AFDとASMlibを使用するLinux OSに固有の設定トピック

ASMlibブロックサイズ

ASMlibは、オプションのASM管理ライブラリおよび関連ユーティリティです。その主な価値は、LUNまたはNFSベースのファイルにASMリソースとして人間が判読可能なラベルを付けることです。

ASMlibの最近のバージョンでは、Logical Blocks Per Physical Block Exponent（LBPPBE）というLUNパラメータが検出されています。最近まで、この値はONTAP SCSIターゲットによって報告されていませんでした。4KBのブロックサイズが推奨されることを示す値が返されるようになりました。これはブロックサイズの定義ではありませんが、LBPPBEを使用するアプリケーションにとって、特定のサイズのI/Oがより効率的に処理される可能性があることを示唆しています。ただし、ASMlibはLBPPBEをブロックサイズとして解釈し、ASMデバイスの作成時にASMヘッダーを永続的にスタンプします。

このプロセスは、さまざまな方法でアップグレードや移行で原因の問題を引き起こす可能性があります。すべては、同じASMディスクグループにブロックサイズの異なるASMlibデバイスを混在させることができないことが原因です。

たとえば、古いアレイでは通常、LBPPBE値が0と報告されているか、この値がまったく報告されていませんでした。ASMlibはこれを512バイトのブロックサイズと解釈します。新しいアレイは、4KBのブロックサイズと解釈されます。512バイトと4KBのデバイスを同じASMディスクグループに混在させることはできません。これにより、2つのアレイのLUNを使用してASMディスクグループのサイズを拡張したり、ASMを移行ツールとして活用したりすることができなくなります。それ以外の場合、RMANでは、512バイトのブロックサイズのASMディスクグループと4KBのブロックサイズのASMディスクグループの間でファイルを複製できないことがあります。

推奨される解決策は、ASMlibにパッチを適用することです。OracleのバグIDは13999609で、パッチはoracleasm-support-2.1.8-1以降に存在します。このパッチを適用すると、ユーザーはパラメータを設定できます。 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 終了： `true` を参照してください `/etc/sysconfig/oracleasm` 構成ファイルこれにより、ASMlibはLBPPBEパラメータを使用できなくなります。つまり、新しいアレイ上のLUNが512バイトのブロックデバイスとして認識されるようになります。

このオプションを使用しても、以前にASMlibによってスタンプされたLUNのブロックサイズは変更されません。たとえば、ブロック数が512バイトのASMディスクグループを、ブロック数が4KBと報告される新しいストレージシステムに移行する必要がある場合は、オプション `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 新しいLUNがASMlibでスタンプされる前に設定する必要があります。 デバイスがoracleasmによってすでにスタンプされている場合は、新しいブロックサイズで再スタンプする前に再フォーマットする必要があります。まず、デバイスの設定を解除します。 `oracleasm deletedisk`をクリックし、デバイスの最初の1GBを消去します。 `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`。最後に、デバイスが以前にパーティション分割されていた場合は、 `kpartx` 古いパーティションを削除するか、単にOSを再起動するためのコマンド。

ASMlibにパッチを適用できない場合は、ASMlibを構成から削除できます。この変更はシステムの停止を伴うため、ASMディスクのスタンプを解除し、 `asm_diskstring` パラメータが正しく設定されている。ただし、この変更ではデータの移行は必要ありません。

ASMフィルタドライブ（AFD）のブロックサイズ

AFDは、ASMlibに代わるオプションのASM管理ライブラリです。ストレージの観点から見ると、ASMlibはASMlibに非常に似ていますが、Oracle以外のI/Oをブロックして、データが破損する可能性のあるユーザーまたはアプリケーションのエラーの可能性を減らすなどの追加機能が含まれています。

デバイスのブロックサイズ

ASMlibと同様に、AFDもLUNパラメータLogical Blocks Per Physical Block Exponent（LBPPBE）を読み取り、デフォルトでは論理ブロックサイズではなく物理ブロックサイズを使用します。

ASMデバイスがすでに512バイトのブロックデバイスとしてフォーマットされている既存の構成にAFDを追加すると、問題が発生する可能性があります。AFDドライバはLUNを4Kデバイスとして認識し、ASMラベルと物理デバイスの不一致が原因でアクセスできなくなります。同様に、512バイトと4KBのデバイスを同じASMディスクグループに混在させることはできないため、移行も影響を受けます。これにより、2つのアレイのLUNを使用してASMディスクグループのサイズを拡張したり、ASMを移行ツールとして活用したりすることができなくなります。それ以外の場合、RMANでは、512バイトのブロックサイズのASMディスクグループと4KBのブロックサイズのASMディスクグループの間でファイルを複製できないことがあります。

解決策はシンプルです- AFDには、論理ブロックサイズと物理ブロックサイズのどちらを使用するかを制御するパラメータが含まれています。これは、システム上のすべてのデバイスに影響を与えるグローバルパラメータです。AFDで強制的に論理ブロックサイズを使用するには、 `options oracleafd oracleafd_use_logical_block_size=1` を参照してください `/etc/modprobe.d/oracleafd.conf` ファイル。

マルチパステンソウサイス

最近のLinuxカーネルの変更では、マルチパスデバイスに送信されるI/Oサイズ制限が適用されますが、AFDではこれらの制限が適用されません。その後I/Oが拒否され、LUNパスがオフラインになります。その結果、Oracle Gridのインストール、ASMの設定、データベースの作成ができなくなります。

解決策では、ONTAP LUNのmultipath.confファイルに最大転送長を手動で指定します。

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

現在問題が存在しない場合でも、AFDを使用して将来のLinuxアップグレードで予期せず原因の問題が発生しないようにする場合は、このパラメータを設定する必要があります。

Microsoft Windows の場合

ONTAPを使用したMicrosoft Windows上のOracleデータベースの構成に関するトピック

NFS

Oracleでは、Direct NFSクライアントでのMicrosoft Windowsの使用がサポートされています。この機能は、複数の環境にわたるファイルの表示、ボリュームの動的なサイズ変更、安価なIPプロトコルの活用など、NFSの管理上のメリットをもたらします。DNFSを使用してMicrosoft Windowsにデータベースをインストールおよび設定する方法については、Oracleの公式ドキュメントを参照してください。特別なベストプラクティスはありません。

SAN

圧縮効率を最適化するには、NTFSファイルシステムで8K以上の割り当て単位を使用するようにしてください。一般にデフォルトである4Kの割り当て単位を使用すると、圧縮効率が低下します。

Solaris の場合

Solaris OSに固有の構成に関するトピック

Solaris NFSのマウントオプション

次の表に、単一インスタンスのSolaris NFSのマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1], roto=tcp, timeo=600,rsize=262144,wsize=262144`
制御ファイル データファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,llock,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`

の使用 `llock` ストレージシステムでロックを取得および解放する際のレイテンシを排除することで、お客様の環境のパフォーマンスが劇的に向上することが実証されています。多数のサーバが同じファイルシステムをマウントするように構成され、Oracleがこれらのデータベースをマウントするよう構成されている環境では、このオプションの使用に注意してください。これは非常に珍しい構成ですが、少数のお客様によって使用されています。インスタンスが誤って2回目に開始された場合、Oracleは外部サーバ上のロックファイルを検出できないため、データが破損する可能性があります。NFSロックは、NFSバージョン3のように保護を提供するものではなく、推奨されるだけです。

なぜなら、 `llock` および `forcedirectio` パラメータは相互に排他的です。次のことが重要です。 `filesystemio_options=setall` は、 `init.ora` ファイルを作成して `directio` を使用します。このパラメータを指定しないと、ホストOSのバッファキャッシュが使用され、パフォーマンスが低下する可能性があります。

次の表に、Solaris NFSのマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,noac`
制御ファイル データ・ファイル REDO ログ	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,forcedirectio`
CRS /投票	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,forcedirectio`
専用 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid`
共有 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac,suid`

シングルインスタンスとRACマウントオプションの主な違いは、 `noac` および `forcedirectio` をマウントオプションに移動します。このオプションを使用するとホストOSのキャッシングが無効になるため、データの状態について、RACクラスタ内のすべてのインスタンスが一貫した情報を認識できるようになります。ただし、 `init.ora` パラメータ `filesystemio_options=setall` ホストのキャッシングを無効にした場合と同じ効果がありますが、引き続きを使用する必要があります。 `noac` および `forcedirectio`。

理由 `actimeo=0` 共有の場合は必須です `ORACLE_HOME` を導入すると、Oracleパスワードファイルやspfileなどのファイルの整合性が維持されます。RACクラスタ内の各インスタンスに専用の `ORACLE_HOME`、このパラメータは必須ではありません。

Solaris UFSのマウントオプション

NetAppでは、ロギングマウントオプションを使用して、Solarisホストがクラッシュしたり**FC**接続が中断したりした場合にデータの整合性が維持されるようにすることを強く推奨しています。ロギングマウントオプションを使用すると、Snapshotバックアップのユーザビリティも維持されます。

Solaris ZFS

最適なパフォーマンスを実現するには、Solaris ZFSをインストールして慎重に設定する必要があります。

mvector

Solaris 11では、大規模なI/O処理の処理方法が変更され、SANストレージアレイのパフォーマンスに重大な問題が発生する可能性がありました。この問題は、NetApp追跡バグレポート630173「Solaris 11 ZFSパフォーマンスの回帰」で文書化されています。

これはONTAPのバグではありません。これはSolarisの障害であり、Solarisの障害7199305および7082975で追跡されます。

使用しているSolaris 11のバージョンが影響を受けるかどうかを確認するには、Oracleサポートを参照してください。また、より小さい値に変更して回避策をテストすることもできます `zfs_mvector_max_size`。

これを行うには、ルートとして次のコマンドを実行します。

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

この変更によって予期しない問題が発生した場合は、次のコマンドをrootとして実行することで簡単に元に戻すことができます。

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

カーネル

信頼性の高いZFSパフォーマンスを実現するには、LUNのアライメントの問題に対してSolarisカーネルにパッチを適用する必要があります。この修正は、Solaris 10のパッチ147440-19とSolaris 11のSRU 10.5で導入されました。ZFSではSolaris 10以降のみを使用してください。

LUN構成

LUNを設定するには、次の手順を実行します。

1. タイプがのLUNを作成します。 `solaris`。
2. で指定された適切なHost Utility Kit（HUK）をインストールします。 "ネットアップの Interoperability Matrix Tool （ IMT ）"。
3. HUKに記載されている手順に正確に従ってください。基本的な手順は以下のとおりですが、 "最新のドキュメント" を参照してください手順。
   a. を実行します `host_config` を更新するユーティリティ `sd.conf/sdd.conf` ファイル。これにより、SCSIドライブがONTAP LUNを正しく検出できるようになります。
   b. の指示に従ってください。 `host_config` Multipath Input/Output（MPIO；マルチパス入出力）を有効にするユーティリティ。
   c. リブートします。この手順は、システム全体で変更が認識されるようにするために必要です。
4. LUNをパーティショニングし、適切にアライメントされていることを確認します。アライメントを直接テ

ストして確認する方法については、「付録B：WAFLアライメントの検証」を参照してください。

zpools

zpoolは、の手順を実行したあとに作成する必要があります。 "LUNの設定" が実行されます。手順を正しく実行しないと、I/Oのアライメントが原因でパフォーマンスが大幅に低下する可能性があります。ONTAPのパフォーマンスを最適化するには、I/Oがドライブの4Kの境界にアライメントされている必要があります。zpoolに作成されるファイルシステムは、というパラメータで制御される実効ブロックサイズを使用します。 `ashift`（コマンドを実行すると表示できます） `zdb -C`。

の値 `ashift` デフォルトは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。

zpoolを作成したら、の値を確認します。 `ashift` 次に進む前に、値が12以外の場合は、LUNが正しく検出されていません。zpoolを削除し、関連するHost Utilitiesのドキュメントに記載された手順をすべて正しく実行したことを確認してから、zpoolを再作成します。

zpoolとSolaris LDOM

Solaris LDOMには、I/Oアライメントが正しいことを確認するための追加の要件があります。LUNは4Kデバイスとして適切に検出されますが、LDOM上の仮想vdskデバイスはI/Oドメインの設定を継承しません。このLUNに基づくvdskは、デフォルトで512バイトブロックに戻ります。

追加の構成ファイルが必要です。まず、追加の設定オプションを有効にするために、個 々 のLDOMにOracleのバグ15824910のパッチを適用する必要があります。このパッチは、現在使用されているすべてのバージョンのSolarisに移植されています。LDOMにパッチを適用すると、適切にアライメントされた新しいLUNを設定できるようになります。手順は次のとおりです。

1. 新しいzpoolで使用するLUNを特定します。この例では、c2d1デバイスです。

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. ZFSプールに使用するデバイスのVDCインスタンスを取得します。

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. 編集 `/platform/sun4v/kernel/drv/vdc.conf`：

```
block-size-list="1:4096";
```

つまり、デバイスインスタンス1には4096のブロックサイズが割り当てられます。

追加の例として、vdskインスタンス1~6を4Kブロックサイズに設定する必要があり、`/etc/path_to_inst` 読み取り値は次のとおりです。

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. 決勝戦 `vdc.conf` ファイルには以下が含まれている必要があります

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

注意
vdc.confを設定してvdskを作成したら、LDOMをリブートする必要があります。この手順は避けられません。ブロックサイズの変更はリブート後にのみ有効になります。zpoolの設定に進み、前述のようにashiftが12に正しく設定されていることを確認します。

ZFSインテントログ（ZIL）

通常'ZFSインテントログ(ZIL)を別のデバイスに配置する理由はありませんログはメインプールとスペースを共有できます。ZILを別 々 に使用する主な用途は、最新のストレージアレイには書き込みキャッシュ機能がない物理ドライブを使用する場合です。

ロバイアス

を設定します `logbias` Oracleデータをホストするzfsファイルシステムのパラメータ。

```
zfs set logbias=throughput <filesystem>
```

このパラメータを使用すると、全体的な書き込みレベルが低下します。デフォルトでは、書き込まれたデータはまずZILにコミットされ、次にメインのストレージプールにコミットされます。このアプローチは、SSDベースのZILデバイスとメインストレージプール用の回転式メディアを含む、プレーンドライブ構成を使用する構成に適しています。これは、利用可能な最も低レイテンシのメディア上の単一のI/Oトランザクションでコミットを実行できるためです。

独自のキャッシュ機能を備えた最新のストレージアレイを使用する場合は、通常、このアプローチは必要ありません。まれに、レイテンシの影響を受けやすい大量のランダム書き込みで構成されるワークロードのように、単一のトランザクションで書き込みをログにコミットした方が望ましい場合があります。ログに記録されたデータは最終的にメインのストレージプールに書き込まれ、書き込みアクティビティが2倍になるため、ライトアンプリフィケーションという結果になります。

ダイレクトI/O

Oracle製品を含む多くのアプリケーションでは、ダイレクトI/Oを有効にすることでホストのバッファキャッシュをバイパスできます。ZFSファイルシステムでは、この方法は想定どおりに機能しません。ホストのバッファキャッシュはバイパスされますが、ZFS自体はデータのキャッシュを継続します。I/Oがストレージシステムに到達しているかどうか、またはI/OがOS内にローカルにキャッシュされているかどうかを予測することが困難であるため、FIOやSIOなどのツールを使用してパフォーマンステストを実行すると、誤った結果になる可能性があります。また、このような総合的なテストを使用してZFSと他のファイルシステムのパフォーマンスを比較することも非常に困難になります。実際のユーザワークロードでは、ファイルシステムのパフォーマンスにほとんど違いはありません。

複数のzpool

ZFSベースのデータのSnapshotベースのバックアップ、リストア、クローニング、アーカイブは、zpoolレベルで実行する必要があり、通常は複数のzpoolが必要です。zpoolはLVMディスクグループに似ており、同じルールを使用して設定する必要があります。たとえば、データベースのレイアウトには、データファイルが配置されているのが最適です。 `zpool1` およびにあるアーカイブログ、制御ファイル、REDOログ `zpool2`。このアプローチでは、データベースがホットバックアップモードに設定された標準のホットバックアップに続いて、 `zpool1`。次に、データベースがホットバックアップモードから削除され、ログアーカイブが強制的に実行され、 `zpool2` が作成されます。リストア処理では、zfsファイルシステムをアンマウントし、zpoolを完全にオフラインにしてから、SnapRestoreのリストア処理を実行する必要があります。その後、zpoolをオンラ

インに戻してデータベースをリカバリできます。

ファイルシステムオプション

Oracleパラメータ `filesystemio_options` ZFSでは動作が異なります。状況 `setall` または `directio` を使用します。書き込み処理は同期でOSのバッファキャッシュをバイパスしますが、読み取りはZFSによってバッファされます。この場合、I/OがZFSキャッシュによって代行受信されて処理されることがあるため、ストレージのレイテンシと総I/Oが想定よりも低くなるため、パフォーマンス分析が困難になります。

ASA r2 システムによるホスト構成

AIX の場合

ASA r2 ONTAPを搭載した IBM AIX 上の Oracle データベースの構成トピック。

次の条件を満たす場合、AIX は Oracle データベースのホスティング用にNetApp ASA r2 でサポートされます。

- Oracle を同時 I/O 用に適切に構成します。
- サポートされている SAN プロトコル (FC/iSCSI/NVMe) を使用します。
- ASA r2 でONTAP 9.16.x 以降を実行します。

同時**I/O**

ASA r2 を搭載した IBM AIX で最適なパフォーマンスを実現するには、同時 I/O を使用する必要があります。同時 I/O がない場合、AIX はシリアル化されたアトミック I/O を実行するため、大きなオーバーヘッドが発生し、パフォーマンスが制限される可能性があります。

当初、 NetAppは `cio` ファイルシステム上で同時 I/O を強制するマウントオプションがありましたが、このプロセスには欠点があり、現在は必要ありません。AIX 5.2 および Oracle 10gR1 の導入以降、AIX 上の Oracle は、ファイル システム全体で同時 I/O を強制するのではなく、同時 I/O 用に個々のファイルを開くことができます。

同時I/Oを有効にする最適な方法は、 `init.ora` パラメータ `filesystemio_options` 終了： `setall`。これにより、Oracleが特定のファイルを開いて同時I/Oで使用できるようになります。

cio をマウント オプションとして使用すると、同時 I/O の使用が強制され、悪影響が生じる可能性があります。たとえば、同時 I/O を強制すると、ファイル システムでの先読みが無効になり、ファイルのコピーやテープ バックアップの実行など、Oracle データベース ソフトウェアの外部で発生する I/O のパフォーマンスが低下する可能性があります。さらに、Oracle GoldenGate や SAP BR*Tools などの製品は、特定のバージョンの Oracle では cio マウント オプションの使用と互換性がありません。

- NetAppの推奨事項*：
- を使用しないでください `cio` ファイルシステムレベルのマウントオプション。代わりに、を使用して同時I/Oを有効にします。 `filesystemio_options=setall`。
- のみを使用してください `cio` 設定できない場合はマウントオプション `filesystemio_options=setall`。

ASA r2 は NAS をサポートしていないため、AIX 上のすべての Oracle デプロイメントではブロック プロトコルを使用する必要があります。

AIX JFS / JFS2のマウントオプション

次の表に、AIX JFS / JFS2のマウントオプションを示します。

ファイルタイプ	マウントオプション
ADRホーム	デフォルト値です
制御ファイル	デフォルト値です
データファイル	デフォルト値です
Redo logs	デフォルト値です
ORACLE_HOME を参照してください	デフォルト値です

AIXを使用する前に `hdisk` データベースを含むあらゆる環境のデバイスでは、パラメータをチェックします `queue_depth`。このパラメータはHBAのキュー深度ではなく、個々のSCSIキュー深度に関係します。 `hdisk device`。ASA r2 LUNの設定方法に応じて、 `queue_depth` 良好なパフォーマンスを得るには低すぎる可能性があります。テストの結果、最適値は 64 であることが分かりました。

HP-UX

ASA r2 ONTAPを搭載した HP-UX 上の Oracle データベースの構成トピック。

次の条件を満たす場合、HP-UX は Oracle データベースのホスティング用にNetApp ASA r2 でサポートされます。

- ONTAPバージョンは 9.16.x 以降です。
- SAN プロトコル (FC/iSCSI/NVMe) を使用します。NAS はASA r2 ではサポートされていません。
- HP-UX 固有のマウントおよび I/O チューニングのベスト プラクティスを適用します。

HP-UX VxFSマウントオプション

Oracleバイナリをホストするファイルシステムには、次のマウントオプションを使用します。

```
delaylog,nodatainlog
```

データファイル、Redoログ、アーカイブログ、制御ファイルが格納されているファイルシステムで、HP-UXのバージョンが同時I/Oをサポートしていない場合は、次のマウントオプションを使用します。

```
nodatainlog,mincache=direct,convosync=direct
```

同時I/Oがサポートされている場合（VxFS 5.0.1以降、またはServiceGuard Storage Management Suiteを使用

している場合）は、データファイル、Redoログ、アーカイブログ、および制御ファイルを格納しているファイルシステムで次のマウントオプションを使用します。

```
delaylog,cio
```

> パラメータ `db_file_multiblock_read_count` VxFS環境では特に重要です。Oracleでは、特に指示がないかぎり、Oracle 10g R1以降ではこのパラメータを未設定のままにすることを推奨しています。Oracleの8KBブロックサイズの場合、デフォルトは128です。このパラメータの値が強制的に16以下になる場合は、 `convosync=direct` シーケンシャルI/Oのパフォーマンスが低下する可能性があるため、マウントオプションを使用します。この手順は、パフォーマンスの他の側面に損傷を与えるため、 `db_file_multiblock_read_count` デフォルト値から変更する必要があります。

Linux の場合

ASA r2 ONTAPを搭載した Linux OS に固有の設定トピック。

> Linux (Oracle Linux、RHEL、SUSE) は、Oracle データベース用のASA r2 でサポートされています。SAN プロトコルを使用し、マルチパスを正しく構成し、ASM および I/O チューニングに関する Oracle のベスト プラクティスを適用します。

I/Oスケジューラ

Linuxカーネルでは、ブロックデバイスへのI/Oのスケジュール方法を低レベルで制御できます。デフォルト値はLinuxのディストリビューションによって大きく異なります。テストでは、通常はDeadlineが最良の結果を提供することが示されていますが、場合によってはNOOPがわずかに改善されています。パフォーマンスの違いはごくわずかですが、データベース構成から最大限のパフォーマンスを引き出す必要がある場合は、両方のオプションをテストしてください。CFQは多くの構成でデフォルトであり、データベースワークロードのパフォーマンスに重大な問題があることが実証されています。

I/Oスケジューラの設定手順については、該当するLinuxベンダーのドキュメントを参照してください。

マルチパス

一部のお客様では、マルチパスデーモンがシステムで実行されていなかったために、ネットワーク停止中にクラッシュが発生しました。最近のバージョンのLinuxでは、OSとマルチパスデーモンのインストールプロセスによって、これらのOSがこの問題に対して脆弱なままになる可能性があります。パッケージは正しくインストールされていますが、再起動後の自動起動が設定されていません。

たとえば、RHEL 9.7 のマルチパス デーモンのデフォルトは次のようになります。

```
[root@host1 ~]# systemctl list-unit-files --type=service | grep multipathd
multipathd.service                          disabled
```

これを修正するには、次のコマンドを使用します。

```
[root@host1 ~]# systemctl enable multipathd.service
[root@host1 ~]# systemctl list-unit-files --type=service | grep multipathd
multipathd.service                          enabled
```

キューの深さ

I/O ボトルネックを回避するために、SAN デバイスに適切なキュー深度を設定します。Linux のデフォルトのキュー深度は 128 に設定されていることが多く、Oracle データベースでパフォーマンスの問題が発生する可能性があります。キューの深さを高く設定しすぎると、過剰な I/O キューが発生し、レイテンシが増加し、スループットが低下する可能性があります。設定値が低すぎると、未処理の I/O 要求の数が制限され、全体的なパフォーマンスが低下する可能性があります。キュー深度 64 は、 ASA r2 上の Oracle データベース ワークロードの開始点として適切であることが多いですが、特定のワークロード特性とパフォーマンス テストに基づいて調整する必要がある場合があります。

ASMミラーリング

ASM ミラーリングでは、 ASM が問題を認識して代替の障害グループに切り替えるために、 Linux マルチパス設定の変更が必要になる場合があります。ONTAP 上のほとんどの ASM 構成では、外部冗長性が使用されます。つまり、データ保護は外部アレイによって提供され、 ASM はデータをミラーリングしません。一部のサイトでは、通常の冗長性を備えた ASM を使用して、通常は異なるサイト間で双方向ミラーリングを提供しています。

アクティブ / アクティブマルチパスをサポートするASA r2 システムの場合、これらのマルチパス設定を調整する必要があります。すべてのパスがアクティブで負荷分散されているため、無期限のキューイングは必要ありません。代わりに、マルチパス パラメータではパフォーマンスと迅速なフェイルバックを優先する必要があります。この動作は ASM ミラーリングにとって重要です。ASM が代替 LUN で I/O を再試行するには、I/O 障害を受信する必要があるためです。I/O が無期限にキューに入れられると、ASM はフェイルオーバーをトリガーできません。

Linuxで次のパラメータを設定します。 `multipath.conf` ASMミラーリングで使用されるASM LUNのファイル：

```
polling_interval 5
no_path_retry 24
failback immediate
path_grouping_policy multibus
path_selector "service-time 0"
```

これらの設定により、ASMデバイスに120秒のタイムアウトが作成されます。タイムアウトは、 `polling_interval` * `no_path_retry` 秒として。状況によっては正確な値の調整が必要になる場合がありますが、ほとんどの場合は120秒のタイムアウトで十分です。具体的には、コントローラのテイクオーバーまたはギブバックが120秒以内に実行され、I/Oエラーが発生しないようにしてください。この場合、障害グループはオフラインになります。

A下限 `no_path_retry` この値を指定すると、ASMが代替障害グループに切り替えるのに必要な時間を短縮できますが、これにより、コントローラのテイクオーバーなどのメンテナンス作業中に不要なフェイルオーバーが発生するリスクも高まります。ASMミラーリングの状態を注意深く監視することで、このリスクを軽減できます。不要なフェイルオーバーが発生した場合、再同期が比較的短時間で実行されると、ミラーを迅速に再同期できます。追加情報については、使用しているOracleソフトウェアのバージョンに対応するASM高速

ミラー再同期に関するOracleのマニュアルを参照してください。

Linuxのxfs、ext3、ext4のマウントオプション

* NetApp は、デフォルトのマウント オプションの使用を推奨します*。LUN 上にファイル システムを作成するときは、適切なアライメントを確保します。

ASMlib/AFD（ASMフィルタドライバ）

ASA r2 ONTAPで AFD と ASMlib を使用する Linux OS に固有の構成トピック。

ASMlibブロックサイズ

ASMlib は、オプションの ASM 管理ライブラリと関連ユーティリティです。その主な価値は、人間が判読できるラベルを使用して LUN を ASM リソースとしてスタンプできる機能です。

ASMlibの最近のバージョンでは、Logical Blocks Per Physical Block Exponent（LBPPBE）というLUNパラメータが検出されています。最近まで、この値はONTAP SCSIターゲットによって報告されていませんでした。4KBのブロックサイズが推奨されることを示す値が返されるようになりました。これはブロックサイズの定義ではありませんが、LBPPBEを使用するアプリケーションにとって、特定のサイズのI/Oがより効率的に処理される可能性があることを示唆しています。ただし、ASMlibはLBPPBEをブロックサイズとして解釈し、ASMデバイスの作成時にASMヘッダーを永続的にスタンプします。

このプロセスは、さまざまな方法でアップグレードや移行で原因の問題を引き起こす可能性があります。すべては、同じASMディスクグループにブロックサイズの異なるASMlibデバイスを混在させることができないことが原因です。

たとえば、古いアレイでは通常、LBPPBE値が0と報告されているか、この値がまったく報告されていませんでした。ASMlibはこれを512バイトのブロックサイズと解釈します。新しいアレイは、4KBのブロックサイズと解釈されます。512バイトと4KBのデバイスを同じASMディスクグループに混在させることはできません。これにより、2つのアレイのLUNを使用してASMディスクグループのサイズを拡張したり、ASMを移行ツールとして活用したりすることができなくなります。それ以外の場合、RMANでは、512バイトのブロックサイズのASMディスクグループと4KBのブロックサイズのASMディスクグループの間でファイルを複製できないことがあります。

推奨される解決策は、ASMlibにパッチを適用することです。OracleのバグIDは13999609で、パッチはoracleasm-support-2.1.8-1以降に存在します。このパッチを適用すると、ユーザーはパラメータを設定できます。 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 終了： `true` を参照してください `/etc/sysconfig/oracleasm` 構成ファイルこれにより、ASMlibはLBPPBEパラメータを使用できなくなります。つまり、新しいアレイ上のLUNが512バイトのブロックデバイスとして認識されるようになります。

このオプションを使用しても、以前にASMlibによってスタンプされたLUNのブロックサイズは変更されません。たとえば、ブロック数が512バイトのASMディスクグループを、ブロック数が4KBと報告される新しいストレージシステムに移行する必要がある場合は、オプション `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 新しいLUNがASMlibでスタンプされる前に設定する必要があります。 デバイスがoracleasmによってすでにスタンプされている場合は、新しいブロックサイズで再スタンプする前に再フォーマットする必要があります。まず、デバイスの設定を解除します。 `oracleasm deletedisk`をクリックし、デバイスの最初の`1GB`を消去します。 `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`。最後に、デバイスが以前にパーティション分割されていた場合は、 `kpartx` 古いパーティションを削除するか、単にOSを再起動するためのコマンド。

ASMlibにパッチを適用できない場合は、ASMlibを構成から削除できます。この変更はシステムの停止を伴うため、ASMディスクのスタンプを解除し、 `asm_diskstring` パラメータが正しく設定されている。ただし、この変更ではデータの移行は必要ありません。

ASMフィルタドライブ（AFD）のブロックサイズ

AFDは、ASMlibに代わるオプションのASM管理ライブラリです。ストレージの観点から見ると、ASMlibはASMlibに非常に似ていますが、Oracle以外のI/Oをブロックして、データが破損する可能性のあるユーザーまたはアプリケーションのエラーの可能性を減らすなどの追加機能が含まれています。

デバイスのブロックサイズ

ASMlibと同様に、AFDもLUNパラメータLogical Blocks Per Physical Block Exponent（LBPPBE）を読み取り、デフォルトでは論理ブロックサイズではなく物理ブロックサイズを使用します。

ASMデバイスがすでに512バイトのブロックデバイスとしてフォーマットされている既存の構成にAFDを追加すると、問題が発生する可能性があります。AFDドライバはLUNを4Kデバイスとして認識し、ASMラベルと物理デバイスの不一致が原因でアクセスできなくなります。同様に、512バイトと4KBのデバイスを同じASMディスクグループに混在させることはできないため、移行も影響を受けます。これにより、2つのアレイのLUNを使用してASMディスクグループのサイズを拡張したり、ASMを移行ツールとして活用したりすることができなくなります。それ以外の場合、RMANでは、512バイトのブロックサイズのASMディスクグループと4KBのブロックサイズのASMディスクグループの間でファイルを複製できないことがあります。

解決策はシンプルです- AFDには、論理ブロックサイズと物理ブロックサイズのどちらを使用するかを制御するパラメータが含まれています。これは、システム上のすべてのデバイスに影響を与えるグローバルパラメータです。AFDで強制的に論理ブロックサイズを使用するには、 `options oracleafd oracleafd_use_logical_block_size=1` を参照してください `/etc/modprobe.d/oracleafd.conf` ファイル。

マルチパステンソウサイス

最近のLinuxカーネルの変更では、マルチパスデバイスに送信されるI/Oサイズ制限が適用されますが、AFDではこれらの制限が適用されません。その後I/Oが拒否され、LUNパスがオフラインになります。その結果、Oracle Gridのインストール、ASMの設定、データベースの作成ができなくなります。

解決策では、ONTAP LUNのmultipath.confファイルに最大転送長を手動で指定します。

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

現在問題が存在しない場合でも、AFDを使用して将来のLinuxアップグレードで予期せず原因の問題が発生しないようにする場合は、このパラメータを設定する必要があります。

Microsoft Windows の場合

ASA r2 ONTAPを使用した Microsoft Windows 上の Oracle データベースの構成トピック。

SAN

圧縮効率を最適化するには、NTFSファイルシステムで8K以上の割り当て単位を使用するようにしてください。一般にデフォルトである4Kの割り当て単位を使用すると、圧縮効率が低下します。

Solaris の場合

ASA r2 ONTAPを搭載した Solaris OS に固有の構成トピック。

Solaris UFSのマウントオプション

NetAppでは、ロギングマウントオプションを使用して、Solarisホストがクラッシュしたり FC接続が中断したりした場合にデータの整合性が維持されるようにすることを強く推奨しています。ロギングマウントオプションを使用すると、Snapshotバックアップのユーザビリティも維持されます。

Solaris ZFS

最適なパフォーマンスを実現するには、Solaris ZFSをインストールして慎重に設定する必要があります。

mvector

Solaris 11では、大規模なI/O処理の処理方法が変更され、SANストレージアレイのパフォーマンスに重大な問題が発生する可能性がありました。この問題は、NetApp追跡バグレポート630173「Solaris 11 ZFSパフォーマンスの回帰」で文書化されています。

これはONTAPのバグではありません。これはSolarisの障害であり、Solarisの障害7199305および7082975で追跡されます。

使用しているSolaris 11のバージョンが影響を受けるかどうかを確認するには、Oracleサポートを参照してください。また、より小さい値に変更して回避策をテストすることもできます `zfs_mvector_max_size`。

これを行うには、ルートとして次のコマンドを実行します。

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

この変更によって予期しない問題が発生した場合は、次のコマンドをrootとして実行することで簡単に元に戻すことができます。

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

カーネル

信頼性の高いZFSパフォーマンスを実現するには、LUNのアライメントの問題に対してSolarisカーネルにパッ

チを適用する必要があります。この修正は、Solaris 10のパッチ147440-19とSolaris 11のSRU 10.5で導入されました。ZFSではSolaris 10以降のみを使用してください。

LUN構成

LUNを設定するには、次の手順を実行します。

1. タイプがのLUNを作成します。 `solaris`。
2. で指定された適切なHost Utility Kit（HUK）をインストールします。 "ネットアップの Interoperability Matrix Tool （ IMT ）"。
3. HUKに記載されている手順に正確に従ってください。基本的な手順は以下のとおりですが、 "最新のドキュメント" を参照してください手順。
   a. を実行します `host_config` を更新するユーティリティ `sd.conf/sdd.conf` ファイル。これにより、SCSIドライブがONTAP LUNを正しく検出できるようになります。
   b. の指示に従ってください。 `host_config` Multipath Input/Output（MPIO；マルチパス入出力）を有効にするユーティリティ。
   c. リブートします。この手順は、システム全体で変更が認識されるようにするために必要です。
4. LUNをパーティショニングし、適切にアライメントされていることを確認します。アライメントを直接テストして確認する方法については、「付録B：WAFLアライメントの検証」を参照してください。

zpools

zpoolは、の手順を実行したあとに作成する必要があります。 "LUNの設定" が実行されます。手順を正しく実行しないと、I/Oのアライメントが原因でパフォーマンスが大幅に低下する可能性があります。ONTAPのパフォーマンスを最適化するには、I/Oがドライブの4Kの境界にアライメントされている必要があります。zpoolに作成されるファイルシステムは、というパラメータで制御される実効ブロックサイズを使用します。 `ashift`（コマンドを実行すると表示できます） `zdb -C`。

の値 `ashift` デフォルトは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。

zpoolを作成したら、の値を確認します。 `ashift` 次に進む前に、値が12以外の場合は、LUNが正しく検出されていません。zpoolを削除し、関連するHost Utilitiesのドキュメントに記載された手順をすべて正しく実行したことを確認してから、zpoolを再作成します。

zpoolとSolaris LDOM

Solaris LDOMには、I/Oアライメントが正しいことを確認するための追加の要件があります。LUNは4Kデバイスとして適切に検出されますが、LDOM上の仮想vdskデバイスはI/Oドメインの設定を継承しません。このLUNに基づくvdskは、デフォルトで512バイトブロックに戻ります。

追加の構成ファイルが必要です。まず、追加の設定オプションを有効にするために、個 々 のLDOMにOracleのバグ15824910のパッチを適用する必要があります。このパッチは、現在使用されているすべてのバージョンのSolarisに移植されています。LDOMにパッチを適用すると、適切にアライメントされた新しいLUNを設定できるようになります。手順は次のとおりです。

1. 新しいzpoolで使用するLUNを特定します。この例では、c2d1デバイスです。

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. ZFSプールに使用するデバイスのVDCインスタンスを取得します。

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. 編集 `/platform/sun4v/kernel/drv/vdc.conf`：

```
block-size-list="1:4096";
```

つまり、デバイスインスタンス1には4096のブロックサイズが割り当てられます。

追加の例として、vdskインスタンス1~6を4Kブロックサイズに設定する必要があり、`/etc/path_to_inst` 読み取り値は次のとおりです。

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. 決勝戦 `vdc.conf` ファイルには以下が含まれている必要があります

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

vdc.confを設定してvdskを作成したら、LDOMをリブートする必要があります。この手順は避けられません。ブロックサイズの変更はリブート後にのみ有効になります。zpoolの設定に進み、前述のようにashiftが12に正しく設定されていることを確認します。

ZFSインテントログ（ZIL）

通常'ZFSインテントログ(ZIL)を別のデバイスに配置する理由はありませんログはメインプールとスペースを共有できます。ZILを別 々 に使用する主な用途は、最新のストレージアレイには書き込みキャッシュ機能がない物理ドライブを使用する場合です。

ロバイアス

を設定します `logbias` Oracleデータをホストする**ZFS**ファイルシステムのパラメータ。

```
zfs set logbias=throughput <filesystem>
```

このパラメータを使用すると、全体的な書き込みレベルが低下します。デフォルトでは、書き込まれたデータはまずZILにコミットされ、次にメインのストレージプールにコミットされます。このアプローチは、SSDベースのZILデバイスとメインストレージプール用の回転式メディアを含む、プレーンドライブ構成を使用する構成に適しています。これは、利用可能な最も低レイテンシのメディア上の単一のI/Oトランザクションでコミットを実行できるためです。

独自のキャッシュ機能を備えた最新のストレージアレイを使用する場合は、通常、このアプローチは必要ありません。まれに、レイテンシの影響を受けやすい大量のランダム書き込みで構成されるワークロードのように、単一のトランザクションで書き込みをログにコミットした方が望ましい場合があります。ログに記録されたデータは最終的にメインのストレージプールに書き込まれ、書き込みアクティビティが2倍になるため、ライトアンプリフィケーションという結果になります。

ダイレクト**I/O**

Oracle製品を含む多くのアプリケーションでは、ダイレクトI/Oを有効にすることでホストのバッファキャッシュをバイパスできます。ZFSファイルシステムでは、この方法は想定どおりに機能しません。ホストのバッファキャッシュはバイパスされますが、ZFS自体はデータのキャッシュを継続します。I/Oがストレージシステムに到達しているかどうか、またはI/OがOS内にローカルにキャッシュされているかどうかを予測することが困難であるため、FIOやSIOなどのツールを使用してパフォーマンステストを実行すると、誤った結果にな

る可能性があります。また、このような総合的なテストを使用してZFSと他のファイルシステムのパフォーマンスを比較することも非常に困難になります。実際のユーザワークロードでは、ファイルシステムのパフォーマンスにほとんど違いはありません。

複数の**zpool**

ZFSベースのデータのSnapshotベースのバックアップ、リストア、クローニング、アーカイブは、zpoolレベルで実行する必要があり、通常は複数のzpoolが必要です。zpoolはLVMディスクグループに似ており、同じルールを使用して設定する必要があります。たとえば、データベースのレイアウトには、データファイルが配置されているのが最適です。 `zpool1` およびにあるアーカイブログ、制御ファイル、REDOログ `zpool2`。このアプローチでは、データベースがホットバックアップモードに設定された標準のホットバックアップに続いて、 `zpool1`。次に、データベースがホットバックアップモードから削除され、ログアーカイブが強制的に実行され、 `zpool2` が作成されます。リストア処理では、zfsファイルシステムをアンマウントし、zpoolを完全にオフラインにしてから、SnapRestoreのリストア処理を実行する必要があります。その後、zpoolをオンラインに戻してデータベースをリカバリできます。

ファイルシステムオプション

Oracleパラメータ `filesystemio_options` ZFSでは動作が異なります。状況 `setall` または `directio` を使用します。書き込み処理は同期でOSのバッファキャッシュをバイパスしますが、読み取りはZFSによってバッファされます。この場合、I/OがZFSキャッシュによって代行受信されて処理されることがあるため、ストレージのレイテンシと総I/Oが想定よりも低くなるため、パフォーマンス分析が困難になります。

AFF/ FASシステム上のネットワーク構成

論理インターフェイス

Oracleデータベースにはストレージへのアクセスが必要です。Logical Interface（LIF；論理インターフェイス）は、Storage Virtual Machine（SVM）をネットワークに接続し、さらにデータベースに接続するネットワーク配管です。各データベースワークロードに十分な帯域幅を確保し、フェイルオーバーによってストレージサービスが失われないようにするには、LIFを適切に設計する必要があります。

このセクションでは、LIFの主な設計原則の概要を説明します。より包括的なドキュメントについては、"ONTAPネットワーク管理に関するドキュメント"。データベースアーキテクチャの他の要素と同様に、Storage Virtual Machine（SVM、CLIではVserver）と論理インターフェイス（LIF）の設計に最適なオプションは、拡張要件とビジネスニーズに大きく依存します。

LIFの戦略を立てる際は、主に次の点を考慮してください。

- *パフォーマンス。*ネットワーク帯域幅は十分か。
- *耐障害性。*設計に単一点障害はありますか？
- *管理性。*ネットワークを無停止で拡張できますか？

これらのトピックは、ホストからスイッチ、ストレージシステムまで、エンドツーエンドの解決策に適用されます。

LIFタイフ

LIFには複数のタイプがあります。 "LIFタイプに関するONTAPのドキュメント" このトピックのより包括的な情報を提供しますが、機能的にはLIFを次のグループに分類できます。

- *クラスタおよびノードの管理LIF。*ストレージクラスタの管理に使用するLIF。
- * SVM管理LIF。* REST APIまたはONTAPI（ZAPIとも呼ばれます）を使用してSVMへのアクセスを許可するインターフェイス。Snapshotの作成やボリュームのサイズ変更などの機能に使用できます。SnapManager for Oracle（SMO）などの製品では、SVM管理LIFにアクセスする必要があります。
- データ**LIF**。 FC、iSCSI、NVMe/FC、NVMe/TCP、NFS、 またはSMB / CIFSデータ。

> ファイアウォールポリシーを `data` 終了： `mgmt` または、HTTP、HTTPS、SSHを許可する別のポリシー。この変更により、NFSデータLIFと別の管理LIFの両方にアクセスするように各ホストを設定する必要がなくなるため、ネットワーク設定が簡易化されます。iSCSIトラフィックと管理トラフィックの両方にIPプロトコルを使用しているにもかかわらず、インターフェイスを設定することはできません。iSCSI環境では、個別の管理LIFが必要です。

SAN LIFの設計

SAN環境でのLIFの設計は、マルチパスという1つの理由で比較的簡単です。最新のSAN実装では、クライアントは複数の独立したネットワークパス経由でデータにアクセスし、アクセスに最適なパス（複数可）を選択できます。その結果、SANクライアントは使用可能な最適なパス間でI/Oの負荷を自動的に分散するため、パフォーマンスに関してはLIFの設計は容易に対処できます。

あるパスが使用できなくなった場合、クライアントは自動的に別のパスを選択します。その結果、設計がシンプルになるため、一般にSAN LIFの管理性が向上します。だからといって、SAN環境の方が常に簡単に管理できるわけではありません。SANストレージには、NFSよりもはるかに複雑な要素が多数あるからです。単純に、SAN LIFの設計が容易であることを意味します。

パフォーマンス

SAN環境でLIFのパフォーマンスを考慮する際に最も重要な考慮事項は、帯域幅です。たとえば、2ノードONTAP AFFクラスタの各ノードに16Gb FCポートを2つ搭載すると、各ノードとの間で最大32Gbの帯域幅を確保できます。

耐障害性

AFFストレージシステムでは、SAN LIFはフェイルオーバーしません。コントローラのフェイルオーバーが原因でSAN LIFに障害が発生すると、クライアントのマルチパスソフトウェアがパスの損失を検出し、I/Oを別のLIFにリダイレクトします。ASAストレージシステムでは、LIFは短時間でフェイルオーバーされますが、もう一方のコントローラにすでにアクティブなパスがあるためIOが中断されることはありません。フェイルオーバープロセスは、定義されたすべてのポートでホストアクセスをリストアするために実行されます。

管理性

NFS環境では、クラスタ内でのボリュームの再配置にLIFの移行が伴うことが多いため、LIFの移行ははるかに一般的なタスクです。SAN環境でHAペア内でボリュームを再配置しても、LIFを移行する必要はありません。ボリュームの移動が完了すると、ONTAPはパスの変更をSANに通知し、SANクライアントは自動的に再最適化します。SANを使用したLIFの移行は、主に物理ハードウェアの大幅な変更に関連しています。たとえば、コントローラの無停止アップグレードが必要な場合は、SAN LIFを新しいハードウェアに移行します。FCポートで障害が検出された場合は、LIFを未使用のポートに移行できます。

設計上の推奨事項

NetAppの推奨事項は次のとおりです。

- 必要以上の数のパスを作成しないでください。パスの数が多すぎると管理全体が複雑になり、一部のホストでのパスのフェイルオーバーで原因の問題が発生する可能性があります。さらに、一部のホストでは、SANブートなどの構成でパスが予期せず制限されます。
- ごく少数の構成では、LUNへのパスが4つ以上必要です。LUNにパスをアドバタイズするノードが3つ以上あると、LUNを所有するノードとそのHAパートナーに障害が発生した場合、LUNをホストしているアグリゲートにアクセスできなくなるため、その価値には制限があります。このような状況では、プライマリHAペア以外のノードにパスを作成しても役に立ちません。
- 参照可能なLUNパスの数はFCゾーンに含めるポートを選択することで管理できますが、一般には、ターゲットとなるポイントをすべてFCゾーンに含め、LUNの可視性をONTAPレベルで制御する方が簡単です。
- ONTAP 8.3以降では、選択的LUNマッピング（SLM）機能がデフォルトです。SLMを使用すると、新しいLUNはすべて、基盤となるアグリゲートを所有するノードとノードのHAパートナーから自動的に通知されます。これにより、ポートのアクセス性を制限するためにポートセットを作成したりゾーニングを設定したりする必要がなくなります。各LUNは、最適なパフォーマンスと耐障害性の両方を実現するために必要な最小限のノードで利用できます。
- LUNを2台のコントローラの外部に移行する必要がある場合は、 `lun mapping add-reporting-nodes` コマンドを実行して、新しいノードでLUNがアドバタイズされるようにします。これにより、LUNの移行用にLUNへの追加のSANパスが作成されます。ただし、新しいパスを使用するには、ホストで検出処理を実行する必要があります。
- 間接トラフィックを過度に気にしないでください。I/Oが大量に発生する環境ではレイテンシがマイクロ秒単位で重要になるため、間接トラフィックは避けることを推奨しますが、一般的なワークロードではパフォーマンスに目に見える影響はごくわずかです。

NFS LIFの設計

NFSでは、SANプロトコルとは異なり、データへの複数のパスを定義する機能に制限があります。NFSv4に対するParallel NFS（pNFS）拡張ではこの制限に対応していますが、イーサネットの速度が100GB以上に達しているため、パスを追加する価値があることはほとんどありません。

パフォーマンスと耐障害性

SAN LIFのパフォーマンスを測定することは、主にすべてのプライマリパスの合計帯域幅を計算することですが、NFS LIFのパフォーマンスを判断するには、正確なネットワーク構成を詳しく調べる必要があります。たとえば、2つの10Gbポートを物理ポートとして構成することも、Link Aggregation Control Protocol（LACP）インターフェイスグループとして構成することもできます。インターフェイスグループとして設定されている場合は、複数のロードバランシングポリシーを使用できます。ロードバランシングポリシーの動作は、トラフィックがスイッチングされるかルーティングされるかによって異なります。最後に、Oracle Direct NFS（dNFS）は、現時点ではどのOS NFSクライアントにも存在しないロードバランシング設定を提供します。

SANプロトコルとは異なり、NFSファイルシステムにはプロトコルレイヤでの耐障害性が必要です。たとえば、LUNは常にマルチパスを有効にして設定されるため、ストレージシステムではFCプロトコルを使用する複数の冗長チャネルを使用できます。一方NFSファイルシステムは、物理レイヤでのみ保護できる単一のTCP/IPチャネルの可用性に依存します。このような理由から、ポートフェイルオーバーやLACPポートアグリゲーションなどのオプションが用意されています。

NFS環境では、パフォーマンスと耐障害性の両方がネットワークプロトコルレイヤで提供されます。その結果、両方のトピックが絡み合っており、一緒に議論する必要があります。

ポートグループへの**LIF**のバインド

LIFをポートグループにバインドするには、LIFのIPアドレスを物理ポートのグループに関連付けます。物理ポートを1つに集約する主な方法はLACPです。LACPのフォールトトレランス機能は非常に簡単です。LACPグループ内の各ポートは監視され、障害が発生した場合はポートグループから削除されます。ただし、パフォーマンスに関してLACPがどのように機能するかについては、多くの誤解があります。

- LACPでは、エンドポイントと一致するようにスイッチで設定する必要はありません。たとえば、ONTAPにIPベースのロードバランシングを設定し、スイッチにMACベースのロードバランシングを使用することができます。
- LACP接続を使用する各エンドポイントは、パケット送信ポートを個別に選択できますが、受信に使用するポートは選択できません。これは、ONTAPから特定の宛先へのトラフィックが特定のポートに結び付けられ、リターントラフィックが別のインターフェイスに到達する可能性があることを意味します。ただし、これは原因の問題ではありません。
- LACPでは、常にトラフィックが均等に分散されるわけではありません。多数のNFSクライアントを含む大規模な環境では、通常はLACPアグリゲーションのすべてのポートが均等に使用されます。ただし、環境内の1つのNFSファイルシステムの帯域幅は、アグリゲーション全体ではなく、1つのポートの帯域幅に制限されます。
- ONTAPではロビンベースのLACPポリシーを使用できますが、スイッチからホストへの接続には対応していません。たとえば、ホストで4ポートのLACPトランクを、ONTAPで4ポートのLACPトランクを使用する構成でも、ファイルシステムの読み取りには1つのポートしか使用できません。ONTAPは4つのポートすべてを介してデータを送信できますが、4つのポートすべてを介してスイッチからホストに送信するスイッチテクノロジは現在使用できません。使用されるのは1つだけです。

多数のデータベースホストで構成される大規模な環境で最も一般的なアプローチは、IPロードバランシングを使用して、適切な数の10Gb（またはそれよりも高速）インターフェイスでLACPアグリゲートを構築する方法です。このアプローチにより、ONTAPはクライアントが十分に存在する限り、すべてのポートを均等に使用できます。LACPトランキングでは負荷が動的に再分散されないため、構成内のクライアント数が少なくなるとロードバランシングが機能しません。

接続が確立されると、特定の方向のトラフィックは1つのポートにのみ配置されます。たとえば、あるデータベースがNFSファイルシステムに対してテーブルのフルスキャンを実行し、接続に4ポートのLACPトランクを使用している場合、データの読み取りには1枚のネットワークインターフェイスカード（NIC）のみが使用されます。このような環境にデータベースサーバが3台しかない場合は、3台すべてが同じポートから読み取りを行い、他の3つのポートはアイドル状態になる可能性があります。

物理ポートへの**LIF**のバインド

物理ポートにLIFをバインドすると、ネットワーク構成をきめ細かく制御できるようになります。これは、ONTAPシステム上の特定のIPアドレスは、一度に1つのネットワークポートにのみ関連付けられるためです。フェイルオーバーグループとフェイルオーバーポリシーを設定することで耐障害性が実現します。

フェイルオーバーポリシーとフェイルオーバーグループ

ネットワーク停止時のLIFの動作は、フェイルオーバーポリシーとフェイルオーバーグループによって制御されます。設定オプションは、ONTAPのバージョンによって変更されました。を参照してください "フェイルオーバーグループとポリシーに関するONTAPのネットワーク管理に関するドキュメント" を参照して、導入するONTAPのバージョンの詳細を確認してください。

ONTAP 8.3以降では、ブロードキャストドメインに基づいてLIFのフェイルオーバーを管理できます。そのため、特定のサブネットにアクセスできるすべてのポートを管理者が定義し、ONTAPが適切なフェイルオーバーLIFを選択できるようにすることができます。このアプローチは一部のお客様にも使用できますが、予測性

がないため、高速ストレージネットワーク環境では制限があります。たとえば、ファイルシステムへの日常的なアクセスに使用する1Gbポートと、データファイルI/Oに使用する10Gbポートの両方を環境に含めることができます。両方のタイプのポートが同じブロードキャストドメインにあると、LIFのフェイルオーバーによって、データファイルI/Oが10Gbポートから1Gbポートに移動される可能性があります。

要約すると、次の方法を検討してください。

1. ユーザ定義のフェイルオーバーグループを設定します。
2. フェイルオーバーグループにストレージフェイルオーバー（SFO）パートナーコントローラのポートを含め、ストレージフェイルオーバー時にLIFがアグリゲートに従って移動するようにします。これにより、間接トラフィックの作成が回避されます。
3. パフォーマンス特性が元のLIFと一致するフェイルオーバーポートを使用します。たとえば、1つの物理10Gbポート上のLIFには、1つの10Gbポートを含むフェイルオーバーグループを含める必要があります。4ポートLACP LIFは、別の4ポートLACP LIFにフェイルオーバーする必要があります。これらのポートは、ブロードキャストドメインに定義されているポートのサブセットになります。
4. SFOパートナーのみにフェイルオーバーポリシーを設定します。これにより、フェイルオーバー時にLIFがアグリゲートに従うようになります。

自動リバート

を設定します `auto-revert` 必要に応じてパラメータを指定する。ほとんどのお客様は、このパラメータを `true` LIFをホームポートにリバートします。ただし、場合によっては、想定外のフェイルオーバーを調査してからLIFをホームポートに戻すように、このパラメータを「false」に設定することもできます。

LIFとボリュームの比率

よくある誤解の1つは、ボリュームとNFS LIFの間には1：1の関係が必要であるということです。この構成は、ボリュームをクラスタ内の任意の場所に移動する際に必要ですが、インターコネクトトラフィックが増えることはありません。ただし、この構成は必須要件ではありません。クラスタ間トラフィックは考慮する必要がありますが、クラスタ間トラフィックが存在するだけでは問題は発生しません。ONTAP用に作成された公開済みのベンチマークの多くには、主に間接I/Oが含まれています。

たとえば、パフォーマンスが重視されるデータベースの数が比較的少なく、合計で40個のボリュームしか必要としないデータベースプロジェクトの場合、ボリューム対LIFの戦略は1：1で、必要なIPアドレスは40個です。これにより、すべてのボリュームを関連付けられたLIFと一緒にクラスタ内の任意の場所に移動でき、トラフィックは常に直接送信されるため、レイテンシのすべてのソースをマイクロ秒レベルでも最小限に抑えることができます。

反対の例として、大規模なホスト環境では、お客様とLIFが1：1の関係にある場合、より簡単に管理できます。時間が経つにつれて、ボリュームを別のノードに移行しなければならない場合があり、間接トラフィックが原因になることがあります。ただし、インターコネクトスイッチのネットワークポートが飽和状態になっていないかぎり、パフォーマンスへの影響は検出されません。懸念がある場合は、ノードを追加して新しいLIFを設定し、次回のメンテナンス時間にホストを更新して、構成から間接トラフィックを取り除くことができます。

TCP/IPおよびイーサネット構成

Oracle on ONTAPをご利用のお客様の多くは、NFS、iSCSI、NVMe/TCPのネットワークプロトコルであるイーサネットを使用しており、特にクラウドを使用しています。

ホストOSの設定

ほとんどのアプリケーションベンダーのドキュメントには、アプリケーションが最適に動作することを確認するためのTCPおよびイーサネットの設定が含まれています。これらの設定は通常、IPベースのストレージパフォーマンスを最適化するのに十分です。

イーサネットフロー制御

このテクノロジを使用すると、クライアントは送信者にデータ転送を一時的に停止するように要求できます。これは通常、受信側が受信データを十分に迅速に処理できないために行われます。一時期、送信者に送信の中止を要求しても、バッファがいっぱいになったために受信者がパケットを破棄するよりも、中断が少なくて済みました。現在OSで使用されているTCPスタックでは、これは当てはまりません。実際、フロー制御は解決するよりも多くの問題を引き起こします。

近年、イーサネットフロー制御に起因するパフォーマンスの問題が増加しています。これは、イーサネットフロー制御が物理レイヤで動作するためです。ネットワーク構成で、任意のホストOSからストレージシステムへのイーサネットフロー制御要求の送信が許可されていると、接続されているすべてのクライアントのI/Oが一時停止します。1台のストレージコントローラで対応するクライアントの数が増えているため、1台以上のクライアントがフロー制御要求を送信する可能性が高くなります。この問題は、OSの仮想化が広範に行われているお客様のサイトで頻繁に発生しています。

NetAppシステム上のNICは、フロー制御要求を受信しないでください。この結果を得る方法は、ネットワークスイッチの製造元によって異なります。ほとんどの場合、イーサネットスイッチのフロー制御は次のように設定できます。 `receive desired` または `receive on`これは、フロー制御要求がストレージコントローラに転送されないことを意味します。それ以外の場合は、ストレージコントローラのネットワーク接続でフロー制御の無効化が許可されないことがあります。このような場合は、ホストサーバ自体またはホストサーバが接続されているスイッチポートのNIC設定に変更して、フロー制御要求を送信しないようにクライアントを設定する必要があります。

* NetAppでは* NetAppストレージコントローラがイーサネットフロー制御パケットを受信しないようにすることを推奨しています。これは通常、コントローラが接続されているスイッチポートを設定することで実行できますが、一部のスイッチハードウェアには制限があり、代わりにクライアント側の変更が必要になる場合があります。

MTUサイス

ジャンボフレームを使用すると、CPUとネットワークのオーバーヘッドが軽減され、1Gbネットワークのパフォーマンスがある程度向上することが示されていますが、通常はそれほど大きなメリットはありません。

* NetAppでは、可能な限りジャンボフレームを実装することを推奨しています。これは、パフォーマンス上のメリットを実現し解決策、将来のニーズにも対応するためです。

10Gbネットワークではジャンボフレームの使用がほぼ必須です。これは、ほとんどの10Gb環境では、ジャンボフレームを使用しないと10Gbに達する前に1秒あたりのパケット数が制限されるためです。ジャンボフレームを使用すると、OS、サーバ、NIC、およびストレージシステムで処理できるパケットの数は少なくても大きいため、TCP / IP処理の効率が向上します。パフォーマンスの向上はNICによって異なりますが、大幅に向上します。

ジャンボフレームの実装では、接続されているすべてのデバイスでジャンボフレームがサポートされている必要があり、MTUサイズがエンドツーエンドで同じである必要があるという誤った考えがよくあります。代わりに、2つのネットワークエンドポイントは、接続を確立するときに、相互に許容可能な最大フレームサイズをネゴシエートします。一般的な環境では、ネットワークスイッチのMTUサイズは9216、NetAppコントロー

ラは9000、クライアントは9000と1514が混在するように設定されています。MTU 9000をサポートできるクライアントはジャンボフレームを使用でき、1514しかサポートできないクライアントは低い値をネゴシエートできます。

完全にスイッチが接続された環境では、この構成に問題が生じることはほとんどありません。ただし、ルーティングされた環境では、中間ルータが強制的にジャンボフレームをフラグメント化しないように注意してください。

- NetAppでは*次の設定を推奨しています。
- ジャンボフレームの使用を推奨しますが、1Gbイーサネット（GbE）の場合は必須ではありません。
- 10GbE以上の速度で最大のパフォーマンスを実現するには、ジャンボフレームが必要です。

TCPパラメータ

TCPタイムスタンプ、選択的確認応答（SACK）、TCPウィンドウスケーリングの3つの設定が誤って設定されることがよくあります。インターネット上の古いドキュメントの多くは、パフォーマンスを向上させるために、これらのパラメータの1つまたは複数を無効にすることを推奨しています。CPU能力がはるかに低く、TCP処理のオーバーヘッドを可能な限り削減できるというメリットが何年も前にあったこの推奨事項には、いくつかのメリットがありました。

ただし、最新のOSでは、これらのTCP機能のいずれかを無効にしても、通常は検出できるメリットはなく、パフォーマンスも低下する可能性があります。特に仮想ネットワーク環境では、パケット損失やネットワーク品質の変化を効率的に処理するためにこれらの機能が必要になるため、パフォーマンスが低下する可能性があります。

* NetAppでは、ホストでTCPタイムスタンプ、SACK、TCPウィンドウスケーリングを有効にすることを推奨しています。現在のOSでは、これら3つのパラメータはすべてデフォルトでオンにする必要があります。

FC SAN構成

Oracleデータベース用にFC SANを構成する主な目的は、日常的なSANのベストプラクティスに従うことです。

これには、ホストとストレージシステムの間のSANに十分な帯域幅があることを確認したり、必要なすべてのデバイス間にすべてのSANパスが存在することを確認したり、FCスイッチベンダーが必要とするFCポート設定を使用してISLの競合を回避したりするなど、一般的な計画方法が含まれます。 SANファブリックを適切に監視します。

ゾーニング

FCゾーンに複数のイニシエータを含めることはできません。このような配置は最初は機能しているように見えるかもしれませんが、最終的にはイニシエータ間のクロストークがパフォーマンスと安定性の妨げになります。

マルチターゲットゾーンは一般に安全とみなされますが、まれに、ベンダーが異なるFCターゲットポートの動作が問題を引き起こすことがあります。たとえば、NetAppとネットアップ以外のストレージアレイのターゲットポートを同じゾーンに配置することは避けてください。また、NetAppストレージシステムとテープデ

バイスを同じゾーンに配置すると、原因の問題が発生する可能性がさらに高くなります。

直接接続ネットワーク

ストレージ管理者は、構成からネットワークスイッチを削除してインフラを簡易化したいと考える場合があります。これは一部のシナリオでサポートされます。

iSCSIとNVMe/TCP

iSCSIまたはNVMe/TCPを使用するホストは、ストレージシステムに直接接続して正常に動作することができます。その理由はパス設定です。2つの異なるストレージコントローラに直接接続すると、データフローが2つの独立したパスになります。パス、ポート、またはコントローラが失われても、他のパスの使用が妨げられることはありません。

NFS

直接接続されたNFSストレージも使用できますが、フェイルオーバーには大きな制限があります。スクリプト作成にはお客様の責任が伴います。

直接接続されたNFSストレージで無停止フェイルオーバーが複雑になるのは、ローカルOSで発生するルーティングが原因です。たとえば、ホストのIPアドレスが192.168.1.1/24で、IPアドレスが192.168.1.50/24のONTAPコントローラに直接接続されているとします。フェールオーバー中、192.168.1.50アドレスはもう一方のコントローラにフェールオーバーでき、ホストが使用できるようになりますが、ホストはそのアドレスの存在をどのように検出しますか。元の192.168.1.1アドレスは、動作中のシステムに接続されていないホストNICに残っています。192.168.1.50宛てのトラフィックは、動作不能なネットワークポートに引き続き送信されます。

2番目のOS NICは19に設定できます。 2.168.1.2およびは、192.168.1.50経由でフェールオーバーされたアドレスと通信できますが、ローカルルーティングテーブルのデフォルトでは、192.168.1.0/24サブネットと通信するために1つの*および1つの*アドレスのみを使用することになります。システム管理者は、失敗したネットワーク接続を検出し、ローカルルーティングテーブルを変更したり、インターフェイスをアップ/ダウンしたりするスクリプトフレームワークを作成できます。正確な手順は、使用しているOSによって異なります。

実際にはNetAppを使用していますが、通常はフェイルオーバー中のIO一時停止が許容されるワークロードのみが対象です。ハードマウントを使用する場合は、一時停止中にIOエラーが発生しないようにしてください。ホスト上のNIC間でIPアドレスを移動するためのフェイルバックまたは手動操作によって、サービスが復元されるまでIOはハングします。

FC直接接続

FCプロトコルを使用してホストをONTAPストレージシステムに直接接続することはできません。その理由はNPIVの使用です。FCネットワークへのONTAP FCポートを識別するWWNは、NPIVと呼ばれる仮想化タイプを使用します。ONTAPシステムに接続されているすべてのデバイスがNPIV WWNを認識できる必要があります。現在、NPIVターゲットをサポートできるホストにインストールできるHBAを提供しているHBAベンダーはありません。

ASA r2 システム上のネットワーク構成

論理インターフェイス

Oracleデータベースにはストレージへのアクセスが必要です。Logical Interface（LIF；

論理インターフェイス）は、Storage Virtual Machine（SVM）をネットワークに接続し、さらにデータベースに接続するネットワーク配管です。各データベースワークロードに十分な帯域幅を確保し、フェイルオーバーによってストレージサービスが失われないようにするには、LIFを適切に設計する必要があります。

このセクションでは、SAN のみの環境向けに最適化されたASA r2 システムの主要な LIF 設計原則の概要を説明します。より詳しい説明については、 "ONTAPネットワーク管理に関するドキュメント"。データベース アーキテクチャの他の側面と同様に、ストレージ仮想マシン (SVM、CLI では vserver と呼ばれる) と論理インターフェイス (LIF) の設計に最適なオプションは、スケーリング要件とビジネス ニーズに大きく依存します。

LIFの戦略を立てる際は、主に次の点を考慮してください。

- *パフォーマンス。*ネットワーク帯域幅は Oracle ワークロードに十分ですか?
- *耐障害性。*設計に単一点障害はありますか？
- *管理性。*ネットワークを無停止で拡張できますか？

これらのトピックは、ホストからスイッチ、ストレージシステムまで、エンドツーエンドの解決策に適用されます。

LIFタイフ

LIFには複数のタイプがあります。 "LIFタイプに関するONTAPのドキュメント" このトピックのより包括的な情報を提供しますが、機能的にはLIFを次のグループに分類できます。

- *クラスタおよびノードの管理LIF。*ストレージクラスタの管理に使用するLIF。
- * SVM管理LIF。* REST APIまたはONTAPI（ZAPIとも呼ばれます）を使用してSVMへのアクセスを許可するインターフェイス。Snapshotの作成やボリュームのサイズ変更などの機能に使用できます。SnapManager for Oracle（SMO）などの製品では、SVM管理LIFにアクセスする必要があります。
- *データ LIF*SAN プロトコル専用のインターフェース: FC、iSCSI、NVMe/FC、NVMe/TCP。NAS プロトコル (NFS、SMB/CIFS) はASA r2 システムではサポートされていません。

> 両方とも IP プロトコルを使用しているにもかかわらず、iSCSI (または NVMe/TCP) と管理トラフィックの両方にインターフェイスを構成することはできません。iSCSI または NVMe/TCP 環境では、個別の管理 LIF が必要です。回復力とパフォーマンスを確保するには、ノードごとにプロトコルごとに複数の SAN データ LIF を構成し、それらを異なる物理ポートとファブリックに分散します。AFF/ FASシステムとは異なり、 ASA r2 では NFS または SMB トラフィックが許可されないため、NAS データ LIF を管理用に再利用するオプションはありません。

SAN LIFの設計

SAN環境でのLIFの設計は、マルチパスという1つの理由で比較的簡単です。最新のSAN実装では、クライアントは複数の独立したネットワークパス経由でデータにアクセスし、アクセスに最適なパス（複数可）を選択できます。その結果、SANクライアントは使用可能な最適なパス間でI/Oの負荷を自動的に分散するため、パフォーマンスに関してはLIFの設計は容易に対処できます。

あるパスが使用できなくなった場合、クライアントは自動的に別のパスを選択します。その結果、設計がシンプルになるため、一般にSAN LIFの管理性が向上します。だからといって、SAN環境の方が常に簡単に管理できるわけではありません。SANストレージには、NFSよりもはるかに複雑な要素が多数あるからです。単純に、SAN LIFの設計が容易であることを意味します。

パフォーマンス

SAN 環境における LIF パフォーマンスに関する最も重要な考慮事項は帯域幅です。たとえば、ノードごとに 2 つの 32Gb FC ポートを備えた 2 ノードのASA r2 クラスタでは、各ノードとの間で最大 64Gb の帯域幅が許可されます。同様に、NVMe/TCP または iSCSI の場合、Oracle ワークロードに十分な 25GbE または 100GbE 接続を確保します。

耐障害性

SAN LIF は、NAS LIF と同じ方法でフェイルオーバーません。ASA r2 システムは、回復力のためにホスト マルチパス (MPIO/ALUA) に依存します。コントローラのフェイルオーバーにより SAN LIF が使用できなくなった場合、クライアントのマルチパス ソフトウェアはパスの損失を検出し、I/O を代替パスにリダイレクトします。ASA r2 は、完全パスの可用性を回復するために、少し遅れて LIF の再配置を実行する場合があります。ただし、パートナー ノードにアクティブ パスがすでに存在するため、これによって I/O が中断されることはありません。定義されたすべてのポートでのホスト アクセスを復元するために、フェイルオーバー プロセスが実行されます。

管理性

HA ペア内でボリュームが再配置される場合、SAN 環境で LIF を移行する必要はありません。これは、ボリュームの移動が完了すると、 ONTAP がパスの変更について SAN に通知を送信し、SAN クライアントが自動的に再最適化を行うためです。SAN を使用した LIF の移行は、主に主要な物理ハードウェアの変更を伴います。たとえば、コントローラの無停止アップグレードが必要な場合、SAN LIF は新しいハードウェアに移行されます。FC ポートに障害があることが判明した場合、LIF を未使用のポートに移行できます。

設計上の推奨事項

NetApp は、 ASA r2 SAN 環境に対して次の推奨事項を示しています。

- 必要以上の数のパスを作成しないでください。パスの数が多すぎると管理全体が複雑になり、一部のホストでのパスのフェイルオーバーで原因の問題が発生する可能性があります。さらに、一部のホストでは、SANブートなどの構成でパスが予期せず制限されます。
- ごく少数の構成では、LUNへのパスが4つ以上必要です。LUNにパスをアドバタイズするノードが3つ以上あると、LUNを所有するノードとそのHAパートナーに障害が発生した場合、LUNをホストしているアグリゲートにアクセスできなくなるため、その価値には制限があります。このような状況では、プライマリHAペア以外のノードにパスを作成しても役に立ちません。
- 参照可能なLUNパスの数はFCゾーンに含めるポートを選択することで管理できますが、一般には、ターゲットとなるポイントをすべてFCゾーンに含め、LUNの可視性をONTAPレベルで制御する方が簡単です。
- デフォルトで有効になっている選択的 LUN マッピング (SLM) 機能を使用します。SLM を使用すると、新しい LUN は、基盤となるアグリゲートを所有するノードとそのノードの HA パートナーから自動的にアドバタイズされます。この配置により、ポート セットを作成したり、ポートのアクセス可能性を制限するためにゾーニングを構成したりする必要がなくなります。各 LUN は、最適なパフォーマンスと回復力の両方に必要な最小限の数のノードで利用できます。
- LUNを2つのコントローラの外部に移行する必要がある場合は、追加のノードを次のように追加できます。 `lun mapping add-reporting-nodes` コマンドを実行して、LUN が新しいノードでアドバタイズされるようにします。これにより、LUN 移行用の LUN への追加の SAN パスが作成されます。ただし、新しいパスを使用するには、ホストで検出操作を実行する必要があります。
- 間接トラフィックを過度に気にしないでください。I/Oが大量に発生する環境ではレイテンシがマイクロ秒単位で重要になるため、間接トラフィックは避けることを推奨しますが、一般的なワークロードではパフォーマンスに目に見える影響はごくわずかです。

TCP/IPおよびイーサネット構成

ASA r2 ONTAP上の Oracle の多くの顧客は、iSCSI および NVMe/TCP のネットワークプロトコルであるイーサネットを使用しています。

ホストOSの設定

ほとんどのアプリケーションベンダーのドキュメントには、アプリケーションが最適に動作することを確認するためのTCPおよびイーサネットの設定が含まれています。これらの設定は通常、IPベースのストレージパフォーマンスを最適化するのに十分です。

イーサネットフロー制御

このテクノロジを使用すると、クライアントは送信者にデータ転送を一時的に停止するように要求できます。これは通常、受信側が受信データを十分に迅速に処理できないために行われます。一時期、送信者に送信の中止を要求しても、バッファがいっぱいになったために受信者がパケットを破棄するよりも、中断が少なくて済みました。現在OSで使用されているTCPスタックでは、これは当てはまりません。実際、フロー制御は解決するよりも多くの問題を引き起こします。

近年、イーサネットフロー制御に起因するパフォーマンスの問題が増加しています。これは、イーサネットフロー制御が物理レイヤで動作するためです。ネットワーク構成で、任意のホストOSからストレージシステムへのイーサネットフロー制御要求の送信が許可されていると、接続されているすべてのクライアントのI/Oが一時停止します。1台のストレージコントローラで対応するクライアントの数が増えているため、1台以上のクライアントがフロー制御要求を送信する可能性が高くなります。この問題は、OSの仮想化が広範に行われているお客様のサイトで頻繁に発生しています。

NetAppシステム上のNICは、フロー制御要求を受信しないでください。この結果を得る方法は、ネットワークスイッチの製造元によって異なります。ほとんどの場合、イーサネットスイッチのフロー制御は次のように設定できます。 `receive desired` または `receive on`これは、フロー制御要求がストレージコントローラに転送されないことを意味します。それ以外の場合は、ストレージコントローラのネットワーク接続でフロー制御の無効化が許可されないことがあります。このような場合は、ホストサーバ自体またはホストサーバが接続されているスイッチポートのNIC設定に変更して、フロー制御要求を送信しないようにクライアントを設定する必要があります。

SAN 専用のASA r2 システムの場合、イーサネットフロー制御の考慮事項は主に iSCSI および NVMe/TCP トラフィックに適用されます。

> * NetApp は* NetApp ASA r2 ストレージ コントローラが Ethernet フロー制御パケットを受信しないようにすることを推奨します。これは通常、コントローラが接続されているスイッチ ポートを設定することによって実行できますが、一部のスイッチ ハードウェアには制限があり、代わりにクライアント側の変更が必要になる場合があります。

MTUサイス

ジャンボフレームを使用すると、CPUとネットワークのオーバーヘッドが軽減され、1Gbネットワークのパフォーマンスがある程度向上することが示されていますが、通常はそれほど大きなメリットはありません。

> * NetAppでは、可能な限りジャンボフレームを実装することを推奨しています。これは、パフォーマンス上のメリットを実現し解決策、将来のニーズにも対応するためです。

SAN 専用のASA r2 システムの場合、ジャンボ フレームはイーサネット ベースの SAN プロトコル (iSCSI お

よび NVMe/TCP) にのみ適用されます。

10Gbネットワークではジャンボフレームの使用がほぼ必須です。これは、ほとんどの10Gb環境では、ジャンボフレームを使用しないと10Gbに達する前に1秒あたりのパケット数が制限されるためです。ジャンボフレームを使用すると、OS、サーバ、NIC、およびストレージシステムで処理できるパケットの数は少なくても大きいため、TCP / IP処理の効率が向上します。パフォーマンスの向上はNICによって異なりますが、大幅に向上します。

ジャンボフレームの実装では、接続されているすべてのデバイスでジャンボフレームがサポートされている必要があり、MTUサイズがエンドツーエンドで同じである必要があるという誤った考えがよくあります。代わりに、2つのネットワークエンドポイントは、接続を確立するときに、相互に許容可能な最大フレームサイズをネゴシエートします。一般的な環境では、ネットワークスイッチのMTUサイズは9216、NetAppコントローラは9000、クライアントは9000と1514が混在するように設定されています。MTU 9000をサポートできるクライアントはジャンボフレームを使用でき、1514しかサポートできないクライアントは低い値をネゴシエートできます。

完全にスイッチが接続された環境では、この構成に問題が生じることはほとんどありません。ただし、ルーティングされた環境では、中間ルータが強制的にジャンボフレームをフラグメント化しないように注意してください。

- NetApp は、 ASA r2 SAN 環境では次のように構成することを推奨します。*
- ジャンボ フレームは 1GbE では望ましいですが必須ではありません。
- 10GbE で最大のパフォーマンスを得るにはジャンボ フレームが必要であり、iSCSI および NVMe/TCP トラフィックの場合はさらに高速です。

TCPパラメータ

TCPタイムスタンプ、選択的確認応答（SACK）、TCPウィンドウスケーリングの3つの設定が誤って設定されることがよくあります。インターネット上の古いドキュメントの多くは、パフォーマンスを向上させるために、これらのパラメータの1つまたは複数を無効にすることを推奨しています。CPU能力がはるかに低く、TCP処理のオーバーヘッドを可能な限り削減できるというメリットが何年も前にあったこの推奨事項には、いくつかのメリットがありました。

ただし、最新のOSでは、これらのTCP機能のいずれかを無効にしても、通常は検出できるメリットはなく、パフォーマンスも低下する可能性があります。特に仮想ネットワーク環境では、パケット損失やネットワーク品質の変化を効率的に処理するためにこれらの機能が必要になるため、パフォーマンスが低下する可能性があります。

* NetAppでは、ホストでTCPタイムスタンプ、SACK、TCPウィンドウスケーリングを有効にすることを推奨しています。現在のOSでは、これら3つのパラメータはすべてデフォルトでオンにする必要があります。

FC SAN構成

ASA r2 システム上の Oracle データベース用に FC SAN を構成するには、主に標準の SAN ベスト プラクティスに従う必要があります。

ASA r2 は SAN のみのワークロード向けに最適化されているため、原則はAFF/ FASと同じで、パフォーマンス、復元力、シンプルさに重点が置かれています。これには、ホストとストレージ システム間の SAN に十分な帯域幅があることを確認する、必要なすべてのデバイス間にすべての SAN パスが存在することを確認す

る、FC スイッチ ベンダーが要求する FC ポート設定を使用する、ISL の競合を回避する、適切な SAN ファブリック モニタリングを使用するなどの一般的な計画対策が含まれます。

ゾーニング

FCゾーンに複数のイニシエータを含めることはできません。このような配置は最初は機能しているように見えるかもしれませんが、最終的にはイニシエータ間のクロストークがパフォーマンスと安定性の妨げになります。

マルチターゲットゾーンは一般に安全とみなされますが、まれに、ベンダーが異なるFCターゲットポートの動作が問題を引き起こすことがあります。たとえば、NetAppとネットアップ以外のストレージアレイのターゲットポートを同じゾーンに配置することは避けてください。また、NetAppストレージシステムとテープデバイスを同じゾーンに配置すると、原因の問題が発生する可能性がさらに高くなります。

- ASA r2 は集約の代わりにストレージ可用性ゾーンを使用しますが、これによって FC ゾーン分割の原則は変更されません。
- マルチパス (MPIO) は依然として主要な復元力メカニズムですが、対称アクティブ / アクティブマルチパスをサポートするASA r2 システムでは、LUN へのすべてのパスがアクティブになり、同時に I/O に使用されます。

直接接続ネットワーク

ストレージ管理者は、構成からネットワークスイッチを削除してインフラを簡易化したいと考える場合があります。これは一部のシナリオでサポートされます。

iSCSIとNVMe/TCP

iSCSI または NVMe/TCP を使用するホストは、 ASA r2 ストレージ システムに直接接続して正常に動作できます。理由はパスです。2 つの異なるストレージ コントローラーに直接接続すると、データ フローに 2 つの独立したパスが作成されます。マルチパスが正しく構成されている場合、パス、ポート、またはコントローラが失われても、他のパスが使用できなくなることはありません。

FC直接接続

FC プロトコルを使用してホストをASA r2 ストレージ システムに直接接続することはできません。理由はAFF/ FASシステムの場合と同じで、NPIV を使用するためです。FC ネットワークへのONTAP FC ポートを識別する WWN は、NPIV と呼ばれる仮想化のタイプを使用します。ONTAPシステムに接続されているすべてのデバイスは、NPIV WWN を認識できる必要があります。現在、NPIV ターゲットをサポートできるホストにインストールできる HBA を提供している HBA ベンダーは存在しません。

AFF / FASシステムでのストレージ構成

FC SAN

LUNアライメント

LUNアライメントとは、基盤となるファイルシステムのレイアウトに合わせてI/Oを最適化することです。

ONTAPシステムでは、ストレージは4KB単位で編成されます。データベースまたはファイルシステムの8KBブロックは、4KBブロック2個に正確にマッピングする必要があります。LUNの構成エラーによってアライメントがいずれかの方向に1KBずれた場合、8KBの各ブロックは、4KBのストレージブロックが2つではなく3つに配置されます。このようにすると、原因によってレイテンシが増加し、ストレージシステム内で実行される原因の追加I/Oが発生します。

アライメントはLVMアーキテクチャにも影響します。論理ボリュームグループ内の物理ボリュームがドライブデバイス全体に定義されている場合（パーティションは作成されません）、LUN上の最初の4KBブロックがストレージシステム上の最初の4KBブロックとアライメントされます。これは正しいアライメントです。パーティションで問題が発生するのは、OSがLUNを使用する開始場所が変わるためです。オフセットが4KB単位でずれているかぎり、LUNはアライメントされます。

Linux環境では、ドライブデバイス全体に論理ボリュームグループを構築します。パーティションが必要な場合は、アライメントをチェックするためにを実行し、各パーティションが8の倍数で開始されていることを確認します `fdisk -u`。つまり、パーティションは8の倍数の512バイトセクター（4KB）から開始されます。

圧縮ブロックのアライメントに関するセクションも参照してください"効率性"。8KBの圧縮ブロックの境界でアライメントされたレイアウトも、4KBの境界でアライメントされます。

ミスアライメントノケイコク

データベースのRedo /トランザクションログでは通常、アライメントされていないI/Oが生成されるため、ONTAPでLUNがミスアライメントされているという警告が原因で誤って表示される可能性があります。

ロギングは、さまざまなサイズの書き込みでログファイルのシーケンシャルライトを実行します。4KBの境界にアライメントされないログ書き込み処理では、次のログ書き込み処理でブロックが完了するため、通常は原因のパフォーマンスの問題は発生しません。その結果、一部の4KBブロックが2つの別 々 の処理で書き込まれていても、ONTAPはほぼすべての書き込みを完全な4KBブロックとして処理できます。

次のようなユーティリティを使用してアライメントを確認します。 `sio` または `dd` 定義されたブロックサイズでI/Oを生成できます。ストレージシステムのI/Oアライメント統計は、 `stats` コマンドを実行しますを参照してください "WAFLアライメントの検証" を参照してください。

Solaris環境ではアライメントがより複雑になります。を参照してください "ONTAP SAN ホスト構成" を参照してください。

注意
Solaris x86環境では、ほとんどの構成に複数のパーティションレイヤがあるため、適切なアライメントにさらに注意してください。Solaris x86パーティションスライスは通常、標準のマスターブートレコードパーティションテーブルの上に存在します。

LUNのサイジングと**LUN**数

Oracleデータベースのパフォーマンスと管理性を最適化するには、最適なLUNサイズと使用するLUNの数を選択することが重要です。

LUNはONTAP上の仮想オブジェクトで、ホストしているアグリゲートのすべてのドライブにわたって配置されます。そのため、LUNはどのサイズを選択してもアグリゲートの潜在的なパフォーマンスを最大限に引き出すため、サイズによるLUNのパフォーマンスへの影響はありません。

便宜上、特定のサイズのLUNを使用したい場合があります。たとえば、データベースを2つの1TB LUNで構成されるLVMまたはOracle ASMディスクグループ上に構築する場合、そのディスクグループは1TB単位で拡張

する必要があります。8個の500GB LUNでディスクグループを構築し、ディスクグループの増分単位を小さくできるようにすることを推奨します。

汎用性に優れた標準LUNサイズを設定すると、管理が複雑になる可能性があるため、推奨されません。たとえば、標準サイズの100GBのLUNは、1TB~2TBのデータベースまたはデータストアの場合に適していますが、サイズが20TBのデータベースまたはデータストアには200個のLUNが必要です。つまり、サーバのリブート時間が長くなり、さまざまなUIで管理するオブジェクトが増え、SnapCenterなどの製品は多くのオブジェクトに対して検出を実行する必要があります。LUNのサイズを大きくすることで、このような問題を回避できます。

- LUNの数は、サイズよりも重要です。
- LUNのサイズは、主に必要なLUN数によって決まります。
- 必要以上の数のLUNを作成することは避けてください。

LUN数

LUNのサイズとは異なり、LUNの数はパフォーマンスに影響します。アプリケーションのパフォーマンスは、多くの場合、SCSIレイヤを介して並列I/Oを実行できるかどうかに左右されます。その結果、2つのLUNの方が単一のLUNよりもパフォーマンスが向上します。Veritas VxVM、Linux LVM2、Oracle ASMなどのLVMを使用すると、並列処理を強化する最も簡単な方法です。

NetAppのお客様は、LUNの数を16個以上に増やすことによるメリットはほとんどありませんが、ランダムI/Oが非常に大きい100% SSD環境のテストでは、最大64個のLUNがさらに向上していることが実証されています。

- NetAppの推奨事項*：

一般に、あらゆるデータベースワークロードのI/Oニーズに対応するには、4~16個のLUNで十分です。LUNを4つ未満にすると、ホストのSCSI実装の制限が原因でパフォーマンスが制限される可能性があります。

LUNノハイチ

ONTAPボリューム内でのデータベースLUNの最適な配置は、主に、さまざまなONTAP機能の使用方法によって異なります。

個のボリューム

ONTAPを初めて導入するお客様と混同される共通点の1つは、FlexVol（一般に単に「ボリューム」と呼ばれる）を使用することです。

ボリュームがLUNではありません。これらの用語は、クラウドプロバイダを含む他の多くのベンダー製品と同義語として使用されています。ONTAPボリュームは、単なる管理コンテナです。単独でデータを提供することも、スペースを占有することもありません。ファイルまたはLUN用のコンテナであり、特に大規模環境で管理性を向上および簡易化するために用意されています。

ボリュームとLUN

関連するLUNは通常、1つのボリュームに同じ場所に配置されます。たとえば、10個のLUNが必要なデータベースでは、通常、10個のLUNすべてが同じボリュームに配置されます。

- LUNとボリュームの比率を1：1（ボリュームごとに1つのLUN）にすることは、正式なベストプラクティスでは*ありません。
- 代わりに、ボリュームをワークロードまたはデータセットのコンテナとみなす必要があります。各ボリュームにLUNを1つだけ配置することも、多数配置することもできます。適切な回答は、管理要件によって異なります。
- LUNを不要な数のボリュームに分散させると、Snapshot処理などの処理でオーバーヘッドやスケジュールに関する追加の問題が発生したり、UIに表示されるオブジェクトの数が多すぎたり、LUNの制限に達する前にプラットフォームのボリューム制限に達したりする可能性があります。

ボリューム、**LUN**、**Snapshot**

Snapshotポリシーとスケジュールは、LUNではなくボリュームに配置されます。10個のLUNで構成されるデータセットでは、これらのLUNが同じボリュームに同じ場所にある場合、Snapshotポリシーは1つだけで済みます。

さらに、1つのボリューム内の特定のデータセットに関連するすべてのLUNを同じ場所に配置することで、アトミックなスナップショット操作が可能になります。たとえば、10個のLUNにあるデータベースや、10個のOSで構成されるVMwareベースのアプリケーション環境を、基盤となるLUNがすべて1つのボリュームに配置されている場合は、1つの整合性のあるオブジェクトとして保護できます。Snapshotが別のボリュームに配置されている場合は、同時にスケジュールされていても、Snapshotが100%同期されている場合とそうでない場合があります。

場合によっては、リカバリ要件のために、関連する一連のLUNを2つのボリュームに分割しなければならないことがあります。たとえば、データベースにデータファイル用のLUNが4つ、ログ用のLUNが2つあるとします。この場合は、4つのLUNを含むデータファイルボリュームと2つのLUNを含むログボリュームが最適なオプションです。その理由は独立した回復可能性です。たとえば、データファイルボリュームを選択して以前の状態にリストアすると、4つのLUNすべてがSnapshotの状態にリバートされ、重要なデータを含むログボリュームには影響はありません。

ボリューム、**LUN**、**SnapMirror**

SnapMirrorのポリシーや処理は、Snapshotの処理と同様に、LUNではなくボリュームに対して実行されます。

関連するLUNを1つのボリュームに同じ場所に配置すると、1つのSnapMirror関係を作成し、1回の更新ですべてのデータを更新できます。スナップショットと同様に、更新もアトミックな操作になります。SnapMirrorデスティネーションには、ソースLUNの単一のポイントインタイムレプリカが保証されます。LUNが複数のボリュームに分散している場合は、レプリカ間で整合性がとれている場合とそうでない場合があります。

ボリューム、**LUN**、**QoS**

QoSは個 々 のLUNに選択して適用できますが、通常はボリュームレベルで設定する方が簡単です。たとえば、特定のESXサーバのゲストが使用するすべてのLUNを1つのボリュームに配置し、ONTAPアダプティブQoSポリシーを適用できます。その結果、すべての環境がTBあたりのIOPS制限を自己拡張できるようになります。

同様に、データベースに100K IOPSが必要で、10個のLUNを使用している場合は、LUNごとに1つずつ10K IOPSの制限を個別に10個設定するよりも、1つのボリュームに100K IOPSの制限を1つ設定する方が簡単です。

マルチボリュームレイアウト

複数のボリュームにLUNを分散すると効果的な場合があります。主な理由は、コントローラのストライピングです。たとえば、HAストレージシステムで単一のデータベースをホストし、各コントローラの処理能力とキャッシュ能力をフルに発揮する必要があるとします。この場合、一般的な設計では、LUNの半分をコントローラ1の1つのボリュームに配置し、残りの半分をコントローラ2の1つのボリュームに配置します。

同様に、コントローラストライピングをロードバランシングに使用することもできます。10個のLUNからなる100個のデータベースをホストするHAシステムは、2台のコントローラそれぞれで5個のLUNのボリュームを各データベースに格納するように設計できます。その結果、追加のデータベースがプロビジョニングされるたびに、各コントローラの対称的なロードが保証されます。

ただし、これらの例では、ボリュームとLUNの比率が1：1である必要はありません。その目標は'関連するLUNをボリューム内に共存させることで'管理性を最適化することです

たとえばコンテナ化では、LUNとボリュームの比率を1：1にすることが理にかなっています。コンテナ化では、各LUNは実際には単一のワークロードに相当し、それぞれを個別に管理する必要があります。このような場合、1:1の比率が最適な場合があります。

LUNのサイズ変更と**LVM**のサイズ変更

SANベースのファイルシステムが容量の上限に達した場合は、次の2つの方法で使用可能なスペースを増やすことができます。

- LUNのサイズを拡張する
- 既存のボリュームグループにLUNを追加し、それに含まれる論理ボリュームを拡張する

LUNのサイズ変更は容量を拡張するためのオプションですが、一般にはOracle ASMなどのLVMを使用することを推奨します。LVMが存在する主な理由の1つは、LUNのサイズ変更を回避することです。LVMでは、複数のLUNが1つの仮想ストレージプールにボンディングされます。このプールから切り分けられた論理ボリュームはLVMで管理されるため、サイズを簡単に変更できます。もう1つの利点は、特定の論理ボリュームを使用可能なすべてのLUNに分散することで、特定のドライブ上のホットスポットを回避できることです。透過的な移行は、通常、ボリュームマネージャを使用して論理ボリュームの基盤となるエクステントを新しいLUNに再配置することで実行できます。

LVMストライピング

LVMストライピングとは、複数のLUNにデータを分散することです。その結果、多くのデータベースのパフォーマンスが大幅に向上します。

フラッシュドライブが登場する以前は、回転式ドライブのパフォーマンス上の制限を克服するためにストライピングが使用されていました。たとえば、OSが1MBの読み取り操作を実行する必要がある場合、1つのドライブからその1MBのデータを読み取るには、1MBがゆっくり転送されるため、多くのドライブヘッドのシークと読み取りが必要になります。この1MBのデータが8つのLUNにストライピングされている場合、OSは8つの128K読み取り処理を並行して問題できるため、1MB転送の完了に必要な時間が短縮されます。

回転式ドライブを使用したストライピングは、I/Oパターンを事前に把握しておく必要があったため、より困難でした。ストライピングが実際のI/Oパターンに合わせて正しく調整されていない場合、ストライピングされた構成ではパフォーマンスが低下する可能性があります。Oracleデータベース、特にオールフラッシュ構成では、ストライピングは設定がはるかに簡単で、パフォーマンスが劇的に向上することが実証されています。

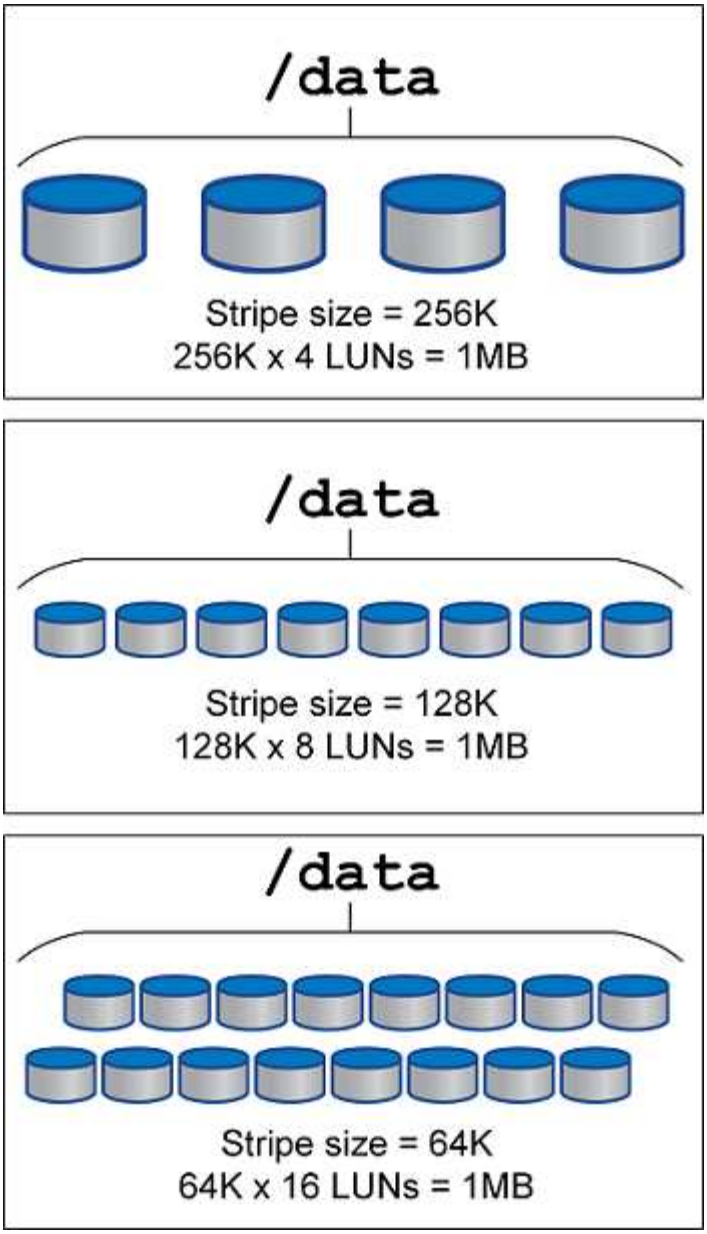
デフォルトではOracle ASMなどの論理ボリュームマネージャがストライプされますが、ネイティブOS LVMは

ストライプされません。その中には、複数のLUNを連結されたデバイスとして結合するものもあります。そのため、データファイルは1つのLUNデバイスにしか存在しません。これにより、ホットスポットが発生します。他のLVM実装では、デフォルトで分散エクステントが使用されます。これはストライピングに似ていますが、粗いです。ボリュームグループ内のLUNはエクステントと呼ばれる大きな部分にスライスされ、通常は数メガバイト単位で測定され、論理ボリュームがそれらのエクステントに分散されます。その結果、ファイルに対するランダムI/OはLUN間で適切に分散されますが、シーケンシャルI/O処理はそれほど効率的ではありません。

高いパフォーマンスを必要とするアプリケーションI/Oは、ほとんどの場合（a）基本ブロックサイズの単位または（b）1メガバイトのいずれかです。

ストライピング構成の主な目的は、シングルファイルI/Oを1つのユニットとして実行し、マルチブロックI/O（サイズは1MB）をストライピングされたボリューム内のすべてのLUNで均等に並列化できるようにすることです。つまり、ストライプ・サイズはデータベース・ブロック・サイズより小さくすることはできず、ストライプ・サイズにLUN数を掛けたサイズは1MBにする必要があります。

次の図に、ストライプサイズと幅の調整に使用できる3つのオプションを示します。LUNの数は、前述のパフォーマンス要件を満たすように選択されますが、いずれの場合も、1つのストライプ内の総データ量は1MBです。



NFS

概要

NetAppは30年以上にわたってエンタープライズクラスのNFSストレージを提供してきましたが、クラウドベースのインフラへの移行が進むにつれ、その使用がますます増えています。その理由は、シンプルさです。

NFSプロトコルには、要件が異なる複数のバージョンが含まれています。ONTAPを使用したNFSの完全な概要構成については、を参照してください。 "TR-4067『NFS on ONTAP Best Practices』"。次のセクションでは、より重要な要件と一般的なユーザーエラーについて説明します。

NFSハアシヨン

オペレーティングシステムNFSクライアントがNetAppでサポートされている必要があります。

- NFSv3は、NFSv3標準に準拠したOSでサポートされます。
- Oracle dNFSクライアントではNFSv3がサポートされています。
- NFSv4は、NFSv4標準に準拠するすべてのOSでサポートされます。
- NFSv4.1およびNFSv4.2では、特定のOSのサポートが必要です。を参照してください "NetApp IMT" サポートされているOSの場合。
- NFSv4.1でのOracle dNFSのサポートには、Oracle 12.2.0.2以降が必要です。

。 "NetAppのサポートマトリックス" NFSv3およびNFSv4の場合、特定のオペレーティングシステムは含まれません。RFCに準拠するすべてのOSが一般的にサポートされています。オンラインのIMTでNFSv3またはNFSv4のサポートを検索する場合は、該当するOSが表示されないため、特定のOSを選択しないでください。すべてのOSは、一般ポリシーで暗黙的にサポートされています。

Linux NFSv3 TCPスロットテーブル

TCPスロットテーブルは、NFSv3でホストバスアダプタ（HBA）のキュー深度に相当します。一度に未処理となることのできるNFS処理の数を制御します。デフォルト値は通常16ですが、最適なパフォーマンスを得るには小さすぎます。逆に、新しいLinuxカーネルでTCPスロットテーブルの上限をNFSサーバが要求でいっぱいになるレベルに自動的に引き上げることができるため、問題が発生します。

パフォーマンスを最適化し、パフォーマンスの問題を回避するには、TCPスロットテーブルを制御するカーネルパラメータを調整します。

を実行します `sysctl -a | grep tcp.*.slot_table` コマンドを実行し、次のパラメータを確認します。

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

すべてのLinuxシステムに `sunrpc.tcp_slot_table_entries`ただし、次のようなものがあります。 `sunrpc.tcp_max_slot_table_entries`。どちらも128に設定する必要があります。

これらのパラメータを設定しないと、パフォーマンスに大きく影響する可能性があります。Linux OSが十分なI/Oを発行していないためにパフォーマンスが制限される場合もあります。一方では、Linux OSが問題で処理できる以上のI/Oを試行すると、I/Oレイテンシが増加します。

ADRとNFS

一部のお客様から、内のデータに対する過剰なI/Oが原因でパフォーマンスの問題が発生すると報告されています。 `ADR` 場所。通常、この問題は、大量のパフォーマンスデータが蓄積されるまで発生しません。過剰なI/Oの理由は不明ですが、Oracleプロセスがターゲットディレクトリを繰り返しスキャンして変更を求めたことが原因と考えられます。

の取り外し `noac` および / または `actimeo=0` マウントオプションを使用すると、ホストOSのキャッシュを実行してストレージI/Oレベルを削減できます。

* NetApp推奨*配置しないこと `ADR` フアイルシステムノテエタ `noac` または `actimeo=0` パフォーマンスの問題が発生する可能性があるからです。分離 `ADR` データを別のマウントポイントに格納（必要な場合）

nfs-rootonlyおよび**mount-rootonly**

ONTAPには、という名前のNFSオプションがあります。 `nfs-rootonly` これにより、サーバが高ポートからのNFSトラフィック接続を受け入れるかどうかが制御されます。セキュリティ対策として、1024未満の送信元ポートを使用してTCP/IP接続を開くことができるのはrootユーザだけです。これは、このようなポートは通常、ユーザプロセスではなくOS用に予約されているためです。この制限により、NFSトラフィックが実際のオペレーティングシステムNFSクライアントからのものであり、NFSクライアントをエミュレートする悪意のあるプロセスではないことを確認できます。Oracle dNFSクライアントはユーザスペースドライバですが、このプロセスはrootとして実行されるため、通常は `nfs-rootonly`。接続は低ポートから行われます。

。 `mount-rootonly` オプションは環境NFSv3のみです。1024より大きいポートからRPCマウント呼び出しを受け入れるかどうかを制御します。dNFSを使用すると、クライアントは再びルートとして実行されるため、1024未満のポートを開くことができます。このパラメータは効果がありません。

NFSバージョン4.0以降でdNFSを使用して接続を開いているプロセスは、rootとして実行されないため、1024以上のポートが必要です。。 `nfs-rootonly` dNFSが接続を完了するには、パラメータをdisabledに設定する必要があります。

状況 `nfs-rootonly` が有効になっている場合、マウントフェーズでdNFS接続を開いているときにハングします。sqlplusの出力は次のようになります。

```
SQL>startup
ORACLE instance started.
Total System Global Area 4294963272 bytes
Fixed Size                  8904776 bytes
Variable Size             822083584 bytes
Database Buffers         3456106496 bytes
Redo Buffers                7868416 bytes
```

パラメータは次のように変更できます。

```
Cluster01::> nfs server modify -nfs-rootonly disabled
```

まれに、nfs-rootonlyとmount-rootonlyの両方をdisabledに変更しなければならないことがあります。サーバが非常に多くのTCP接続を管理している場合、1024未満のポートが使用できない可能性があり、OSはより高いポートを使用するように強制されます。接続を完了するには、これら2つのONTAPパラメータを変更する必要があります。

NFSエクスポートポリシー：**superuser**と**setuid**

OracleバイナリがNFS共有に配置されている場合は、エクスポートポリシーにsuperuser権限とsetuid権限を含める必要があります。

ユーザホームディレクトリなどの汎用ファイルサービスに使用される共有NFSエクスポートでは、通常、root

ユーザが引き下げられます。これは、ファイルシステムをマウントしたホスト上のrootユーザからの要求が、より低い権限を持つ別のユーザとして再マッピングされることを意味します。これは、特定のサーバ上のrootユーザが共有サーバ上のデータにアクセスできないようにすることで、データを保護するのに役立ちます。setuidビットは、共有環境ではセキュリティリスクになることもあります。setuidビットを使用すると、コマンドを呼び出すユーザとは別のユーザとしてプロセスを実行できます。たとえば、rootがsetuidビットを持つシェルスクリプトはrootとして実行されます。そのシェルスクリプトが他のユーザによって変更される可能性がある場合、root以外のユーザはスクリプトを更新することでrootとしてコマンドを問題できます。

Oracleバイナリには、rootが所有するsetuidビットを使用するファイルが含まれます。OracleバイナリがNFS共有にインストールされている場合は、エクスポートポリシーに適切なsuperuser権限とsetuid権限が含まれている必要があります。次の例では、ルールに `allow-suid` 許可します `superuser` （root）システム認証を使用したNFSクライアントのアクセス。

```
Cluster01::> export-policy rule show -vserver vserver1 -policyname orabin
-fields allow-suid,superuser
vserver   policyname ruleindex superuser allow-suid
--------- ---------- --------- --------- ----------
vserver1  orabin     1         sys       true
```

NFSv4 / 4.1構成

ほとんどのアプリケーションで、NFSv3とNFSv4の違いはほとんどありません。通常、アプリケーションI/Oは非常に単純なI/Oであり、NFSv4の高度な機能の一部からあまりメリットが得られません。上位バージョンのNFSは、データベースストレージから見ると「アップグレード」ではなく、機能を追加したNFSのバージョンとみなすべきです。たとえば、Kerberosプライバシーモード（krb5p）のエンドツーエンドのセキュリティが必要な場合は、NFSv4が必要です。

* NetAppでは* NFSv4の機能が必要な場合はNFSv4.1を使用することを推奨します。NFSv4.1では、一部のエッジにおける耐障害性を向上させるために、NFSv4プロトコルの機能がいくつか強化されています。

NFSv4への切り替えは、マウントオプションを単にvers=3からvers=4.1に変更するよりも複雑です。ONTAPを使用したNFSv4設定の詳細（OSの設定に関するガイダンスなど）については、を参照してください。 "TR-4067『NFS on ONTAP』のベストプラクティス"。このTRの以降のセクションでは、NFSv4を使用するための基本的な要件の一部について説明します。

NFSv4ドメイン

NFSv4 / 4.1の設定について詳しくは本ドキュメントでは説明していませんが、よく発生する問題の1つとして、ドメインマッピングの不一致があります。sysadminから見ると、NFSファイルシステムは正常に動作しているように見えますが、アプリケーションからは特定のファイルに対する権限やsetuidに関するエラーが報告されます。場合によっては、管理者は、アプリケーションバイナリのアクセス許可が破損していると誤って判断し、実際の問題がドメイン名であったときにchownまたはchmodコマンドを実行しました。

ONTAP SVMでNFSv4ドメイン名が設定されます。

```
Cluster01::> nfs server show -fields v4-id-domain
vserver   v4-id-domain
--------- ------------
vserver1  my.lab
```

ホストのNFSv4ドメイン名は、 `/etc/idmap.cfg`

```
[root@host1 etc]# head /etc/idmapd.conf
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
Domain = my.lab
```

ドメイン名が一致している必要があります。マッピングされていない場合は、次のようなマッピングエラーがに表示されます。 `/var/log/messages`：

```
Apr 12 11:43:08 host1 nfsidmap[16298]: nss_getpwnam: name 'root@my.lab'
does not map into domain 'default.com'
```

アプリケーションバイナリ（Oracleデータベースバイナリなど）には、rootが所有するsetuidビットのファイルが含まれています。つまり、NFSv4ドメイン名が一致していないとOracleの起動に失敗し、という名前のファイルの所有権または権限に関する警告が表示されます。 `oradism`をクリックします。 `$ORACLE_HOME/bin` ディレクトリ。次のように表示されます。

```
[root@host1 etc]# ls -l /orabin/product/19.3.0.0/dbhome_1/bin/oradism
-rwsr-x--- 1 root oinstall 147848 Apr 17  2019
/orabin/product/19.3.0.0/dbhome_1/bin/oradism
```

所有権がnobodyのファイルが表示される場合は、NFSv4ドメインのマッピングに問題がある可能性があります。

```
[root@host1 bin]# ls -l oradism
-rwsr-x--- 1 nobody oinstall 147848 Apr 17  2019 oradism
```

これを修正するには、 `/etc/idmap.cfg` ファイルをONTAPのv4-id-domain設定に対して作成し、整合性を確保します。設定されていない場合は、必要な変更を行い、 `nfsidmap -c`をクリックし、変更が反映されるまでしばらく待ちます。これで、ファイル所有権が`root`として正しく認識されます。ユーザがを実行しようとした場合 `chown root` NFSドメインの設定が修正される前に、このファイルで次のコマンドを実行する必要があります。 `chown root` をもう一度クリックします

Oracle Direct NFS（dNFS）

Oracleデータベースでは、NFSを2つの方法で使用できます。

まず、オペレーティングシステムの一部であるネイティブのNFSクライアントを使用してマウントされたファイルシステムを使用できます。これはカーネルNFS（kNFS）と呼ばれることもあります。NFSファイルシステムは、NFSファイルシステムを使用する他のアプリケーションとまったく同じように、Oracleデータベースによってマウントされ、使用されます。

2つ目の方法はOracle Direct NFS（dNFS）です。これは、Oracleデータベースソフトウェアに標準を実装したものです。DBAによるOracleデータベースの設定や管理方法は変更されません。ストレージシステム自体に正しい設定があるかぎり、DBAチームやエンドユーザはdNFSを透過的に使用できなければなりません。

dNFS機能を有効にしたデータベースでは、通常のNFSファイルシステムが引き続きマウントされています。データベースが開くと、Oracleデータベースは一連のTCP/IPセッションを開き、NFS操作を直接実行します。

Direct NFS

OracleのDirect NFSの主なメリットは'ホストのNFSクライアントをバイパスして'NFSサーバ上で直接NFSファイル操作を実行することですこれを有効にするには、Oracle Disk Manager（ODM）ライブラリを変更する必要があります。このプロセスの手順については、Oracleのマニュアルを参照してください。

dNFSを使用すると、I/Oが最も効率的な方法で実行されるため、I/Oパフォーマンスが大幅に向上し、ホストとストレージシステムの負荷が軽減されます。

さらに、Oracle dNFSには、ネットワークインターフェイスのマルチパスとフォールトトレランス用の*オプション*が含まれています。たとえば、2つの10Gbインターフェイスをバインドして、20Gbの帯域幅を提供できます。一方のインターフェイスで障害が発生すると、もう一方のインターフェイスでI/Oが再試行されます。全体的な処理はFCマルチパスとほぼ同じです。マルチパスは、1Gbイーサネットが最も一般的な標準であった数年前には一般的でした。ほとんどのOracleワークロードには10Gb NICで十分ですが、必要に応じて10Gb NICをボンディングできます。

dNFSを使用する場合は、Oracleのドキュメント1495104.1に記載されているパッチをすべてインストールしておくことが重要です。パッチをインストールできない場合は、環境を評価して、そのドキュメントに記載されているバグが原因の問題ではないことを確認する必要があります。必要なパッチをインストールできないためにdNFSを使用できない場合があります。

dNFSでラウンドロビン方式の名前解決（DNS、DDNS、NISなど）を使用しないでください。これには、ONTAPで使用できるDNSロードバランシング機能も含まれます。dNFSを使用するOracleデータベースがあるホスト名をIPアドレスに解決した場合、以降の検索でもホスト名が変更されないようにする必要があります。その結果、Oracleデータベースがクラッシュし、データが破損する可能性があります。

dNFSノユウコウカ

Oracle dNFSは、dNFSライブラリを有効にする以外の設定は必要ありませんが（必要なコマンドについてはOracleのマニュアルを参照してください）、dNFSで接続を確立できない場合は、カーネルNFSクライアントにサイレントに戻すことができます。この場合、パフォーマンスに重大な影響を与える可能性があります。

NFSv4.Xで複数のインターフェイス間でdNFSの多重化を使用する場合、または暗号化を使用する場合は、oranfstabファイルを設定する必要があります。構文は非常に厳格です。ファイルに小さなエラーがあると、起動がハングしたり、oranfstabファイルがバイパスされたりする可能性があります。

本書の執筆時点では、最新バージョンのOracle DatabaseであるNFSv4.1では、dNFSマルチパスは機能しませ

ん。プロトコルとしてNFSv4.1を指定するoranfstabファイルでは、特定のエクスポートに対して1つのpathステートメントのみを使用できます。これは、ONTAPがclientIDトランキングをサポートしていないためです。この制限を解決するOracle Databaseパッチは、将来提供される可能性があります。

dNFSが期待どおりに動作することを確認する唯一の方法は、v$dnfsテーブルを照会することです。

以下は、/etcにあるサンプルoranfstabファイルです。これは、oranfstabファイルを配置できる複数の場所の1つです。

```
[root@jfs11 trace]# cat /etc/oranfstab
server: NFSv3test
path: jfs_svmdr-nfs1
path: jfs_svmdr-nfs2
export: /dbf mount: /oradata
export: /logs mount: /logs
nfs_version: NFSv3
```

最初に、指定したファイルシステムでdNFSが動作していることを確認します。

```
SQL> select dirname,nfsversion from v$dnfs_servers;

DIRNAME
-----------------------------------
NFSVERSION
----------------
/logs
NFSv3.0

/dbf
NFSv3.0
```

この出力は、これら2つのファイルシステムでdNFSが使用されていることを示していますが、oranfstabが動作していることを意味するわけではない*。エラーが発生していた場合、dNFSはホストのNFSファイルシステムを自動検出しますが、このコマンドからも同じ出力が表示されることがあります。

マルチパスは次のようにチェックできます。

```
SQL> select svrname,path,ch_id from v$dnfs_channels;

SVRNAME
-----------------------------------
PATH
-----------------------------------
     CH_ID
----------
```

```
NFSv3test
jfs_svmdr-nfs1
         0

NFSv3test
jfs_svmdr-nfs2
         1

SVRNAME
-----------------------------------
PATH
-----------------------------------
     CH_ID
----------

NFSv3test
jfs_svmdr-nfs1
         0

NFSv3test
jfs_svmdr-nfs2

[output truncated]

SVRNAME
-----------------------------------
PATH
-----------------------------------
     CH_ID
----------
NFSv3test
jfs_svmdr-nfs2
         1

NFSv3test
jfs_svmdr-nfs1
         0

SVRNAME
-----------------------------------
PATH
-----------------------------------
     CH_ID
----------

NFSv3test
```

```
jfs_svmdr-nfs2
          1


66 rows selected.
```

これらはdNFSが使用している接続です。SVRNAMEエントリごとに2つのパスとチャネルが表示されます。これは、マルチパスが機能していること、つまりoranfstabファイルが認識されて処理されたことを意味します。

Direct NFSとホストファイルシステムへのアクセス

アプリケーションやユーザのアクティビティが、ホストにマウントされた参照可能なファイルシステムに依存している場合、dNFSを使用すると原因の問題が発生することがあります。これは、dNFSクライアントがホストOSの帯域外でファイルシステムにアクセスするためです。dNFSクライアントは、OSが認識されていなくてもファイルの作成、削除、および変更を行うことができます。

シングルインスタンスデータベースのマウントオプションを使用すると、ファイルおよびディレクトリの属性のキャッシュが有効になり、ディレクトリの内容もキャッシュされます。そのため、dNFSでファイルが作成される可能性があり、OSがディレクトリの内容を再読み取りしてファイルがユーザに表示されるまでに少し時間がかかります。これは通常は問題になりませんが、まれに、SAP BR * Toolsなどのユーティリティで問題が発生することがあります。この場合は、マウントオプションをOracle RACの推奨事項に変更して、問題に対処してください。この変更により、ホストのキャッシュがすべて無効になります。

マウントオプションを変更するのは、（a）dNFSが使用されていて、（b）問題がファイルが参照可能になるまでの遅延が原因で発生した場合のみにしてください。dNFSを使用していない場合は、シングルインスタンスデータベースでOracle RACマウントオプションを使用すると、パフォーマンスが低下します。

通常とは異なる結果になる可能性のあるLinux固有のdNFSの問題については、の"Linux NFSのマウントオプション"注を参照してください `nosharecache`。

NFSのリースとロック

NFSv3はステートレスです。つまり、NFSサーバ（ONTAP）は、どのファイルシステムがマウントされているのか、誰がどのロックが実際に有効であるのかを追跡しません。

ONTAPにはマウントの試行を記録する機能がいくつかあります。そのため、どのクライアントがデータにアクセスしている可能性があるかを把握できます。また、アドバイザリロックが存在する可能性がありますが、その情報が100%完了する保証はありません。NFSクライアントの状態の追跡はNFSv3標準には含まれていないため、この処理を完了できません。

NFSv4のステートフル

一方、NFSv4はステートフルです。NFSv4サーバは、どのクライアントが使用しているファイルシステム、どのファイルが存在するか、どのファイルやファイル領域がロックされているかなどを追跡します。 つまり、状態のデータを最新の状態に保つためには、NFSv4サーバ間で定期的な通信が必要です。

NFSサーバによって管理されている最も重要な状態は、NFSv4ロックとNFSv4リースであり、これらは非常に密接に関連しています。それぞれがそれ自体でどのように機能し、それらが互いにどのように関連しているかを理解する必要があります。

NFSv4ロック

NFSv3では、ロックは推奨されます。NFSクライアントは、「ロックされた」ファイルを変更または削除できます。NFSv3のロックは自動的に期限切れになるわけではなく、削除する必要があります。これは問題を引き起こします。たとえば、クラスタ化されたアプリケーションでNFSv3ロックを作成していて、いずれかのノードで障害が発生した場合は、どうすればよいですか？残りのノードでアプリケーションをコーディングしてロックを解除できますが、それが安全であることをどのようにして確認できますか？「failed」ノードは動作しているが、クラスタの残りのノードと通信していない可能性があります。

NFSv4では、ロックの期間に制限があります。ロックを保持しているクライアントがNFSv4サーバにチェックインし続けるかぎり、他のクライアントはこれらのロックを取得できません。クライアントがNFSv4へのチェックインに失敗すると、最終的にサーバによってロックが取り消され、他のクライアントはロックを要求および取得できます。

NFSv4リース

NFSv4ロックはNFSv4リースに関連付けられます。NFSv4クライアントがNFSv4サーバとの接続を確立すると、リースを取得します。クライアントがロックを取得した場合（ロックにはさまざまな種類があります）、ロックはリースに関連付けられます。

このリースには定義済みのタイムアウトがあります。デフォルトでは、ONTAPはタイムアウト値を30秒に設定します。

```
Cluster01::*> nfs server show -vserver vserver1 -fields v4-lease-seconds

vserver   v4-lease-seconds
--------- ----------------
vserver1  30
```

つまり、NFSv4クライアントはリースを更新するために、30秒ごとにNFSv4サーバにチェックインする必要があります。

リースはすべてのアクティビティによって自動的に更新されるため、クライアントが作業を行っている場合は追加操作を実行する必要はありません。アプリケーションが静かになり、実際の作業を行っていない場合は、代わりに一種のキープアライブ操作(シーケンスと呼ばれる)を実行する必要があります。それは基本的に「私はまだここにいる、私のリースを更新してください」と言っているだけです。

```
 *Question:* What happens if you lose network connectivity for 31 seconds?
```
NFSv3はステートレスです。クライアントからの通信を期待していません。NFSv4はステートフルであり、リース期間が経過するとリースが期限切れになり、ロックが取り消され、ロックされたファイルが他のクライアントから利用可能になります。

NFSv3では、ネットワークケーブルを移動したり、ネットワークスイッチをリブートしたり、設定を変更したりすることができ、問題が発生しないように十分に確認することができます。アプリケーションは通常、ネットワーク接続が再び機能するのを辛抱強く待つだけです。

NFSv4では、作業を完了するまでに30秒かかります（ONTAP内でそのパラメータの値を増やした場合を除く）。それを超えると、リースはタイムアウトになります。通常、この結果、アプリケーションがクラッシュします。

たとえば、Oracleデータベースを使用していて、リースタイムアウトを超えるネットワーク接続（「ネットワークパーティション」と呼ばれることもあります）が失われると、データベースがクラッシュします。

これが発生した場合のOracleアラート・ログの出力例を次に示します

```
2022-10-11T15:52:55.206231-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00202: control file: '/redo0/NTAP/ctrl/control01.ctl'
ORA-27072: File I/O error
Linux-x86_64 Error: 5: Input/output error
Additional information: 4
Additional information: 1
Additional information: 4294967295
2022-10-11T15:52:59.842508-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: '/redo1/NTAP/ctrl/control02.ctl'
ORA-27061: waiting for async I/Os failed
```

syslogを確認すると、次のエラーのいくつかが表示されます。

```
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
```

ログメッセージは通常、アプリケーションがフリーズする以外に、問題の最初の兆候です。通常、ネットワークの停止中は何も表示されません。これは、NFSファイルシステムにアクセスしようとするプロセスとOS自体がブロックされるためです。

エラーは、ネットワークが再び動作可能になると表示されます。上記の例では、接続が再確立されると、OSはロックの再取得を試みましたが、遅すぎました。リースが期限切れになり、ロックが削除されました。その結果、エラーがOracleレイヤまで伝播し、アラートログにメッセージが記録されます。これらのパターンは、データベースのバージョンと構成によって異なる場合があります。

要約すると、NFSv3はネットワークの中断は許容されますが、NFSv4はより機密性が高く、リース期間が定義されます。

30秒のタイムアウトが許容されない場合はどうなりますか。スイッチが再起動されたり、ケーブルが再配置されたりする動的に変化するネットワークを管理していて、その結果、時 々 ネットワークが中断される場合はどうなりますか。リース期間を延長することもできますが、その場合はNFSv4猶予期間の説明が必要です。

NFSv4猶予期間

NFSv3サーバをリブートすると、ほぼ瞬時にIOを処理できるようになります。それはクライアントの状態を

維持することではありませんでした。そのため、ONTAPのテイクオーバー処理はほぼ瞬時に実行されることがよくあります。コントローラがデータの提供を開始する準備ができた時点で、ネットワークにARPを送信し、トポロジの変更を通知します。通常、クライアントはこれをほぼ瞬時に検出し、データの流れを再開します。

ただし、NFSv4では一時停止が発生します。これは、NFSv4がどのように機能するかの一部にすぎません。

> 次のセクションは、ONTAP 9 .15.1の時点で最新のものですが、リースとロックの動作、チューニングオプションはバージョンごとに変更される可能性があります。NFSv4リース/ロックのタイムアウトを調整する必要がある場合は、NetAppサポートで最新情報を確認してください。

NFSv4サーバは、リース、ロック、および誰がどのデータを使用しているかを追跡する必要があります。NFSサーバがパニック状態になってリブートされた場合、または一時的に電力が失われた場合、またはメンテナンス作業中に再起動された場合は、リース/ロックなどのクライアント情報が失われます。サーバは、処理を再開する前に、どのクライアントがどのデータを使用しているかを把握する必要があります。ここで猶予期間が入ります。

NFSv4サーバの電源が突然再投入された場合。再起動すると、IOを再開しようとするクライアントは、基本的に「リース/ロック情報が失われました。ロックを再登録しますか？" これが猶予期間の始まりですONTAPではデフォルトで45秒です。

```
Cluster01::> nfs server show -vserver vserver1 -fields v4-grace-seconds

vserver   v4-grace-seconds
--------- ----------------
vserver1  45
```

その結果、再起動後、すべてのクライアントがリースとロックを再要求する間、コントローラはIOを一時停止します。猶予期間が終了すると、サーバはIO処理を再開します。

この猶予期間は、ネットワークインターフェイスの変更時のリース再生を制御しますが、ストレージフェイルオーバー時のリース再生を制御する2つ目の猶予期間があります `locking.grace_lease_seconds`。これはノードレベルのオプションです。

```
cluster01::> node run [node names or *] options
locking.grace_lease_seconds
```

たとえば、LIFのフェイルオーバーを頻繁に実行する必要があり、猶予期間を短縮する必要がある場合は、を変更します `v4-grace-seconds`。コントローラフェイルオーバー時のIO再開時間を短縮するには、を変更する必要があります `locking.grace_lease_seconds`。

これらの値は、リスクと結果を十分に理解した上で慎重に変更してください。NFSv4.Xでのフェイルオーバー処理や移行処理でのIOの一時停止を完全に回避することはできません。ロック、リース、猶予期間はNFS RFCの一部です。多くのお客様には、フェイルオーバーにかかる時間が短縮されるため、NFSv3を使用することを推奨します。

猶予期間とリース期間が接続されます。前述したように、デフォルトのリースタイムアウトは30秒です。つまり、NFSv4クライアントは少なくとも30秒ごとにサーバにチェックインする必要があります。そうしないと、リースとロックが失われます。この猶予期間はNFSサーバがリース/ロックデータを再構築できるようにするためのもので、デフォルトは45秒です。猶予期間はリース期間よりも長くする必要があります。これにより、リースを30秒以上更新するように設計されたNFSクライアント環境では、再起動後にサーバにチェックインできます。猶予期間を45秒に設定することで、少なくとも30秒ごとにリースを更新することを期待するすべてのクライアントが確実に更新する機会を得ることができます。

30秒のタイムアウトが許容されない場合は、リース期間を延長することもできます。

60秒のネットワーク停止に耐えるためにリースタイムアウトを60秒に延長する場合は、猶予期間も延長する必要があります。つまり、コントローラフェイルオーバー中にIOが一時停止する時間が長くなります。

これは通常は問題ではありません。一般的なユーザはONTAPコントローラを年に1~2回更新するだけで、ハードウェア障害による計画外フェイルオーバーは非常にまれです。また、ネットワークに60秒のネットワーク停止が発生する可能性があり、リースタイムアウトを60秒にする必要がある場合は、まれにストレージシステムのフェイルオーバーに異議を唱えず、61秒の一時停止も発生する可能性があります。ネットワークが60秒以上頻繁に一時停止していることをすでに認識しています。

NFSキャッシング

次のマウントオプションが存在すると、ホストのキャッシュが無効になります。

```
cio, actimeo=0, noac, forcedirectio
```

これらの設定は、ソフトウェアのインストール、パッチ適用、およびバックアップ/リストアの処理速度に重大な悪影響を及ぼす可能性があります。場合によっては、特にクラスタ化されたアプリケーションでは、クラスタ内のすべてのノードにキャッシュの一貫性を提供する必要があるため、これらのオプションが必然的に必要になることがあります。それ以外の場合、顧客はこれらのパラメータを誤って使用し、結果は不要な性能の損傷です。

多くのお客様は、アプリケーションバイナリのインストール時やパッチ適用時に、これらのマウントオプションを一時的に削除しています。インストールまたはパッチ適用プロセス中にターゲットディレクトリを他のプロセスがアクティブに使用していないことをユーザーが確認した場合は、この削除を安全に実行できます。

NFSテンソウサイス

ONTAPでは、デフォルトでNFS I/Oサイズが64Kに制限されています。

ほとんどのアプリケーションとデータベースでランダムI/Oを実行すると、ブロックサイズがはるかに小さくなり、最大64Kよりもはるかに小さくなります。ラージブロックI/Oは通常並列処理されるため、最大64Kも最大帯域幅の確保に制限されるわけではありません。

一部のワークロードでは、最大64Kに制限があります。特に、バックアップ/リカバリ処理やデータベースのフルテーブルスキャンなどのシングルスレッド処理は、実行回数が少なくても大容量のI/Oを実行できるのであれば、より高速かつ効率的に実行できます。ONTAPに最適なI/O処理サイズは256Kです。

特定のONTAP SVMの最大転送サイズは、次のように変更できます。

```
Cluster01::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size
262144
Cluster01::*>
```

ONTAPで許容される最大転送サイズを、現在マウントされているNFSファイルシステムのrsize/wsizeの値より小さくしないでください。これにより、一部のオペレーティングシステムでハングしたり、データが破損したりする可能性があります。たとえば、NFSクライアントのrsize / wsizeが65536に設定されている場合は、クライアント自体が制限されているため、ONTAPの最大転送サイズを65536~1048576の間で調整しても効果はありません。最大転送サイズを65536未満に縮小すると、可用性やデータが損傷する可能性があります。

NVFail

NVFailは、重大なフェイルオーバーシナリオの際に整合性を確保するONTAPの機能です。

データベースは大規模な内部キャッシュを保持するため、ストレージフェイルオーバー時に破損の影響を受けやすくなります。構成全体の健全性に関係なく、重大なイベントによってONTAPフェイルオーバーの強制またはMetroClusterスイッチオーバーの強制が必要になった場合は、以前に確認された変更が実質的に破棄されることがあります。ストレージアレイの内容が時間を遡るようになり、データベースキャッシュの状態がディスク上のデータの状態を反映しなくなります。この不整合により、データが破損します。

キャッシュはアプリケーションレイヤまたはサーバレイヤで実行できます。たとえば、プライマリサイトとリモートサイトの両方でアクティブなサーバを使用するOracle Real Application Cluster（RAC）構成では、Oracle SGA内のデータがキャッシュされます。強制スイッチオーバー処理によってデータが失われると、SGAに格納されているブロックがディスク上のブロックと一致しない可能性があるため、データベースが破損するリスクがあります。

キャッシュの使用は、OSファイルシステムレイヤではあまり明白ではありません。マウントされたNFSファイルシステムのブロックは、OSにキャッシュされる場合があります。または、プライマリサイトにあるLUNに基づくクラスタ化されたファイルシステムをリモートサイトのサーバにマウントして、データをキャッシュすることもできます。このような状況でNVRAMの障害、強制テイクオーバー、強制スイッチオーバーが発生すると、ファイルシステムが破損する可能性があります。

ONTAPは、NVFAILとその関連設定を使用して、このシナリオからデータベースとオペレーティングシステムを保護します。

ASM Reclamation Utility（ASMRU）

インライン圧縮が有効な場合、ONTAPはファイルまたはLUNに書き込まれた初期化済みブロックを効率的に削除します。Oracle ASM Reclamation Utility（ASRU）などのユーティリティは、未使用のASMエクステントにゼロを書き込むことで機能します。

これにより、DBAはデータが削除されたあとにストレージアレイのスペースを再生できます。ONTAPはゼロをインターセプトし、LUNからスペースの割り当てを解除します。ストレージシステム内にデータが書き込ま

れていないため、再生プロセスは非常に高速です。

データベースに関しては、ASMディスクグループには0が含まれているため、LUNのこれらの領域を読み取ると0のストリームが生成されますが、ONTAPはドライブに0を格納しません。代わりに、メタデータが単純に変更され、LUNの初期化された領域がデータの空として内部的にマークされます。

同様の理由から、ゼロのブロックは実際にはストレージアレイ内で書き込みとして処理されないため、初期化されたデータを使用したパフォーマンステストは無効です。

ASRUを使用する場合は、Oracleが推奨するすべてのパッチがインストールされていることを確認してください。

ASA R2システムでのストレージ構成

FC SAN

LUNアライメント

LUNアライメントとは、基盤となるファイルシステムのレイアウトに合わせてI/Oを最適化することです。

ASA r2 システムは、 AFF/ FASと同じONTAPアーキテクチャを使用しますが、構成モデルは簡素化されています。ASA r2 システムは、アグリゲートの代わりにストレージ アベイラビリティ ゾーン (SAZ) を使用しますが、 ONTAP はプラットフォーム間で一貫してブロック レイアウトを管理するため、アライメントの原則は同じままです。ただし、次のASA固有の点に注意してください。

- ASA r2 システムは、すべての LUN に対してアクティブ / アクティブ対称パスを提供するため、アライメント中のパスの非対称性に関する懸念が排除されます。
- ストレージ ユニット (LUN) はデフォルトでシンプロビジョニング。アライメントによってこの動作は変更されません。
- SnapshotリザーブとSnapshotの自動作成削除は、LUN の作成中に設定できます (ONTAP 9.18.1 以降)。

ONTAPシステムでは、ストレージは4KB単位で編成されます。データベースまたはファイルシステムの8KBブロックは、4KBブロック2個に正確にマッピングする必要があります。LUNの構成エラーによってアライメントがいずれかの方向に1KBずれた場合、8KBの各ブロックは、4KBのストレージブロックが2つではなく3つに配置されます。このようにすると、原因によってレイテンシが増加し、ストレージシステム内で実行される原因の追加I/Oが発生します。

アライメントはLVMアーキテクチャにも影響します。論理ボリュームグループ内の物理ボリュームがドライブデバイス全体に定義されている場合（パーティションは作成されません）、LUN上の最初の4KBブロックがストレージシステム上の最初の4KBブロックとアライメントされます。これは正しいアライメントです。パーティションで問題が発生するのは、OSがLUNを使用する開始場所が変わるためです。オフセットが4KB単位でずれているかぎり、LUNはアライメントされます。

Linux環境では、ドライブデバイス全体に論理ボリュームグループを構築します。パーティションが必要な場合は、アライメントをチェックするためにを実行し、各パーティションが8の倍数で開始されていることを確認します `fdisk -u`。つまり、パーティションは8の倍数の512バイトセクター（4KB）から開始されます。

圧縮ブロックのアライメントに関するセクションも参照してください"効率性"。8KBの圧縮ブロックの境界でアライメントされたレイアウトも、4KBの境界でアライメントされます。

ミスアライメントノケイコク

データベースのRedo /トランザクションログでは通常、アライメントされていないI/Oが生成されるため、ONTAPでLUNがミスアライメントされているという警告が原因で誤って表示される可能性があります。

ロギングは、さまざまなサイズの書き込みでログファイルのシーケンシャルライトを実行します。4KBの境界にアライメントされないログ書き込み処理では、次のログ書き込み処理でブロックが完了するため、通常は原因のパフォーマンスの問題は発生しません。その結果、一部の4KBブロックが2つの別 々 の処理で書き込まれていても、ONTAPはほぼすべての書き込みを完全な4KBブロックとして処理できます。

次のようなユーティリティを使用してアライメントを検証します。 `sio` または `dd` 定義されたブロック サイズで I/O を生成できます。ストレージシステムのI/Oアライメント統計は、 `stats` 指示。見る "WAFLアライメントの検証" 詳細についてはこちらをご覧ください。

Solaris環境ではアライメントがより複雑になります。を参照してください "ONTAP SAN ホスト構成" を参照してください。

> Solaris x86環境では、ほとんどの構成に複数のパーティションレイヤがあるため、適切なアライメントにさらに注意してください。Solaris x86パーティションスライスは通常、標準のマスターブートレコードパーティションテーブルの上に存在します。

追加のベストプラクティス:

- HBA ファームウェアと OS 設定をNetApp Interoperability Matrix Tool (IMT) と照合して検証します。
- sanlun ユーティリティを使用して、パスの健全性と配置を確認します。
- Oracle ASM および LVM の場合、アライメントの問題を回避するために、構成ファイル (/etc/lvm/lvm.conf、/etc/sysconfig/oracleasm) が適切に設定されていることを確認してください。

LUNのサイジングと**LUN**数

Oracleデータベースのパフォーマンスと管理性を最適化するには、最適なLUNサイズと使用するLUNの数を選択することが重要です。

LUN は、 ASA r2 システム上のホスティング ストレージ アベイラビリティ ゾーン (SAZ) 内のすべてのドライブに存在するONTAP上の仮想化オブジェクトです。その結果、LUN はどのサイズを選択しても SAZ の潜在的パフォーマンスを最大限に活用するため、LUN のパフォーマンスはそのサイズの影響を受けません。

便宜上、特定のサイズのLUNを使用したい場合があります。たとえば、データベースを2つの1TB LUNで構成されるLVMまたはOracle ASMディスクグループ上に構築する場合、そのディスクグループは1TB単位で拡張する必要があります。8個の500GB LUNでディスクグループを構築し、ディスクグループの増分単位を小さくできるようにすることを推奨します。

汎用性に優れた標準LUNサイズを設定すると、管理が複雑になる可能性があるため、推奨されません。たとえば、標準サイズの100GBのLUNは、1TB~2TBのデータベースまたはデータストアの場合に適していますが、サイズが20TBのデータベースまたはデータストアには200個のLUNが必要です。つまり、サーバのリブート時間が長くなり、さまざまなUIで管理するオブジェクトが増え、SnapCenterなどの製品は多くのオブジェクトに対して検出を実行する必要があります。LUNのサイズを大きくすることで、このような問題を回避できます。

- ASA r2 の考慮事項:*
- ASA r2 の最大 LUN サイズは 128 TB であり、パフォーマンスに影響を与えることなく、より少ない数の

より大きな LUN を使用できます。

- ASA r2 は、集約の代わりにストレージ可用性ゾーン (SAZ) を使用しますが、これによって Oracle ワークロードの LUN サイズ設定ロジックは変更されません。
- シン プロビジョニングはデフォルトで有効になっています。LUN のサイズ変更は中断を伴わず、オフラインにする必要もありません。

LUN数

LUNのサイズとは異なり、LUNの数はパフォーマンスに影響します。アプリケーションのパフォーマンスは、多くの場合、SCSIレイヤを介して並列I/Oを実行できるかどうかに左右されます。その結果、2つのLUNの方が単一のLUNよりもパフォーマンスが向上します。Veritas VxVM、Linux LVM2、Oracle ASMなどのLVMを使用すると、並列処理を強化する最も簡単な方法です。

ASA r2 では、 ONTAP がプラットフォーム間で同様に並列 I/O を処理するため、LUN 数の原則はAFF/ FASと同じままです。ただし、 ASA r2 の SAN 専用アーキテクチャとアクティブ / アクティブ対称パスにより、すべての LUN にわたって一貫したパフォーマンスが保証されます。

NetAppのお客様は、LUNの数を16個以上に増やすことによるメリットはほとんどありませんが、ランダムI/Oが非常に大きい100% SSD環境のテストでは、最大64個のLUNがさらに向上していることが実証されています。

- NetAppの推奨事項*：

一般に、特定の Oracleデータベース ワークロードの I/O ニーズをサポートするには、4 ～ 16 個の LUN で十分です。LUN が 4 つ未満の場合、ホスト SCSI 実装の制限によりパフォーマンスが制限される可能性があります。LUN を 16 個以上に増やしても、極端な場合 (非常に高いランダム I/O SSD ワークロードなど) を除き、パフォーマンスが向上することはほとんどありません。

LUNノハイチ

ASA r2 システム内でのデータベース LUN の最適な配置は、主にさまざまなONTAP機能がどのように使用されるかによって異なります。

ASA r2 システムでは、ストレージ ユニット (LUN または NVMe 名前空間) は、HA ペアのストレージの共通プールとして機能する、ストレージ可用性ゾーン (SAZ) と呼ばれる簡略化されたストレージ レイヤから作成されます。

通常、HA ペアごとにストレージの可用性ゾーン (SAZ) は 1 つだけあります。

ストレージ可用性ゾーン (SAZ)

ASA r2 システムではボリュームは依然として存在しますが、ストレージ ユニットが作成されるときに自動的に作成されます。ストレージ ユニット (LUN または NVMe 名前空間) は、ストレージ可用性ゾーン (SAZ) に自動的に作成されたボリューム内で直接プロビジョニングされます。この設計により、手動でのボリューム管理が不要になり、Oracle データベースなどのブロック ワークロードのプロビジョニングがより直接的かつ合理的になります。

SAZとストレージユニット

関連するストレージ ユニット (LUN または NVMe 名前空間) は通常、単一のストレージ可用性ゾーン (SAZ) 内に共存します。たとえば、10 個のストレージ ユニット (LUN) を必要とするデータベースでは、通常、簡潔さとパフォーマンスのために 10 個のユニットすべてを同じ SAZ に配置します。

- ストレージ ユニットとボリュームの比率を 1:1 にする (つまり、ボリュームごとに 1 つのストレージ ユニット (LUN) を使用する) のが、 ASA r2 のデフォルトの動作です。
- ASA r2 システムに複数の HA ペアがある場合、特定のデータベースのストレージ ユニット (LUN) を複数の SAZ に分散して、コントローラの使用率とパフォーマンスを最適化できます。

FC SAN のコンテキストでは、ここでのストレージ ユニットは LUN を指します。

整合性グループ（CG）、LUN、スナップショット

ASA r2 では、スナップショット ポリシーとスケジュールは、整合性グループ レベルで適用されます。整合性グループ レベルは、調整されたデータ保護のために複数の LUN または NVMe 名前空間をグループ化する論理構造です。10 個の LUN で構成されるデータセットでは、それらの LUN が同じ整合性グループの一部である場合、1 つのスナップショット ポリシーのみが必要になります。

整合性グループは、含まれるすべての LUN にわたってアトミック スナップショット操作を保証します。たとえば、10 個の LUN 上に存在するデータベース、または 10 個の異なる OS で構成される VMware ベースのアプリケーション環境は、基礎となる LUN が同じ整合性グループにグループ化されている場合、単一の一貫性のあるオブジェクトとして保護できます。異なる整合性グループに配置されている場合、スナップショットは、同時にスケジュールされていても、完全に同期されない可能性があります。

場合によっては、回復要件のために、関連する LUN セットを 2 つの異なる整合性グループに分割する必要があります。たとえば、データベースにはデータファイル用の LUN が 4 つ、ログ用の LUN が 2 つある場合があります。この場合、4 つの LUN を持つデータファイル整合性グループと 2 つの LUN を持つログ整合性グループが最適なオプションになる可能性があります。その理由は、独立した回復可能性です。データファイル整合性グループを以前の状態に選択的に復元できるため、4 つの LUN すべてがスナップショットの状態に戻されますが、重要なデータを含むログ整合性グループは影響を受けません。

CG、LUN、 SnapMirror

SnapMirrorポリシーと操作は、スナップショット操作と同様に、LUN ではなく整合性グループで実行されます。

関連する LUN を単一の整合性グループ内に共存させることで、単一のSnapMirror関係を作成し、含まれるすべてのデータを 1 回の更新で更新できます。スナップショットと同様に、更新もアトミック操作になります。SnapMirrorデスティネーションには、ソース LUN の単一のポイントインタイムレプリカが存在することが保証されます。LUN が複数の整合性グループに分散されている場合、レプリカは相互に整合性がある場合とない場合があります。

ASA r2 システムでのSnapMirrorレプリケーションには次の制限があります。

- SnapMirror同期レプリケーションはサポートされていません。
- SnapMirrorアクティブ同期は、2 つのASA r2 システム間でのみサポートされます。
- SnapMirror非同期レプリケーションは、2 つのASA r2 システム間でのみサポートされます。
- SnapMirror非同期レプリケーションは、 ASA r2 システムとASA、 AFF 、 FASシステムまたはクラウド間ではサポートされません。

詳細はこちら "ASA r2 システムでサポートされるSnapMirrorレプリケーション ポリシー"。

CG、LUN、QoS

QoS は個々の LUN に選択的に適用できますが、通常は整合性グループ レベルで設定する方が簡単です。たとえば、特定のESXサーバ内のゲストが使用するすべての LUN を単一の整合性グループに配置し、 ONTAP アダプティブ QoS ポリシーを適用できます。その結果、すべての LUN に適用される、TiB あたりの IOPS 制限が自己スケーリングされます。

同様に、データベースが 100K IOPS を必要とし、10 個の LUN を占有している場合、各 LUN に 1 つずつ、10 個の個別の 10K IOPS 制限を設定するよりも、単一の整合性グループに 1 つの 100K IOPS 制限を設定する方が簡単です。

複数のCGレイアウト

LUN を複数の整合性グループに分散すると有益な場合があります。主な理由はコントローラのストライピングです。たとえば、HA ASA r2 ストレージ システムは、各コントローラの完全な処理およびキャッシュの能力が必要とされる単一の Oracle データベースをホストしている場合があります。この場合、一般的な設計では、LUN の半分をコントローラ 1 の単一の整合性グループに配置し、LUN の残り半分をコントローラ 2 の単一の整合性グループに配置します。

同様に、多数のデータベースをホストする環境では、LUN を複数の整合性グループに分散することで、コントローラーの使用率のバランスを確保できます。たとえば、それぞれ 10 個の LUN を持つ 100 個のデータベースをホストする HA システムでは、データベースごとにコントローラ 1 の整合性グループに 5 個の LUN を割り当て、コントローラ 2 の整合性グループに 5 個の LUN を割り当てることがあります。これにより、追加のデータベースがプロビジョニングされるときに対称的な読み込みが保証されます。

ただし、これらの例ではいずれも LUN と整合性グループの比率が 1:1 ではありません。目標は、関連する LUN を整合性グループに論理的にグループ化することで、管理性を最適化することです。

1:1 の LUN と整合性グループの比率が意味を成す例としては、コンテナ化されたワークロードが挙げられます。コンテナ化されたワークロードでは、各 LUN は実際には個別のスナップショットおよびレプリケーション ポリシーを必要とする単一のワークロードを表す可能性があり、そのため個別に管理する必要があります。このような場合、1:1 の比率が最適な場合があります。

LUNのサイズ変更とLVMのサイズ変更

SAN ベースのファイル システムまたは Oracle ASMディスク グループがASA r2 の容量制限に達した場合、使用可能なスペースを増やすには 2 つのオプションがあります。

- 既存のLUN（ストレージユニット）のサイズを増やす

- 既存のASMディスク グループまたはLVMボリュームグループに新しいLUNを追加し、含まれる論理ボリュームを拡張します。

LUN のサイズ変更はASA r2 でサポートされていますが、通常は Oracle ASM などの論理ボリューム マネージャ (LVM) を使用する方が適切です。LVM が存在する主な理由の 1 つは、LUN のサイズ変更を頻繁に行う必要性を回避することです。LVM を使用すると、複数の LUN が仮想ストレージ プールに結合されます。このプールから切り出された論理ボリュームは、基盤となるストレージ構成に影響を与えることなく簡単にサイズを変更できます。

LVM または ASM を使用することによる追加の利点は次のとおりです。

- パフォーマンスの最適化: 複数の LUN に I/O を分散し、ホットスポットを削減します。
- 柔軟性: 既存のワークロードを中断することなく新しい LUN を追加します。
- 透過的な移行: ASM または LVM は、ホストのダウンタイムなしで、バランス調整や階層化のためにエクステントを新しい LUN に再配置できます。

ASA r2 の主な考慮事項:

- LUN のサイズ変更は、ストレージ可用性ゾーン (SAZ) の容量を使用して、ストレージ VM (SVM) 内のストレージ ユニット レベルで実行されます。
- Oracle の場合、ストライプ化と並列性を維持するために、既存の LUN のサイズを変更するのではなく、LUN を ASM ディスク グループに追加することがベスト プラクティスです。

LVMストライピング

LVMストライピングとは、複数のLUNにデータを分散することです。その結果、多くのデータベースのパフォーマンスが大幅に向上します。

フラッシュドライブが登場する以前は、回転式ドライブのパフォーマンス上の制限を克服するためにストライピングが使用されていました。たとえば、OSが1MBの読み取り操作を実行する必要がある場合、1つのドライブからその1MBのデータを読み取るには、1MBがゆっくり転送されるため、多くのドライブヘッドのシークと読み取りが必要になります。この1MBのデータが8つのLUNにストライピングされている場合、OSは8つの128K読み取り処理を並行して問題できるため、1MB転送の完了に必要な時間が短縮されます。

回転ドライブによるストライピングは、I/O パターンを事前に知っておく必要があったため、より困難でした。ストライピングが実際の I/O パターンに合わせて正しく調整されていない場合、ストライピング構成によってパフォーマンスが低下する可能性があります。Oracle データベース、特にオールフラッシュ ストレージ構成では、ストライピングの構成がはるかに簡単になり、パフォーマンスが劇的に向上することが実証されています。

デフォルトではOracle ASMなどの論理ボリュームマネージャがストライプされますが、ネイティブOS LVMはストライプされません。その中には、複数のLUNを連結されたデバイスとして結合するものもあります。そのため、データファイルは1つのLUNデバイスにしか存在しません。これにより、ホットスポットが発生します。他のLVM実装では、デフォルトで分散エクステントが使用されます。これはストライピングに似ていますが、粗いです。ボリュームグループ内のLUNはエクステントと呼ばれる大きな部分にスライスされ、通常は数メガバイト単位で測定され、論理ボリュームがそれらのエクステントに分散されます。その結果、ファイルに対するランダムI/OはLUN間で適切に分散されますが、シーケンシャルI/O処理はそれほど効率的ではありません。

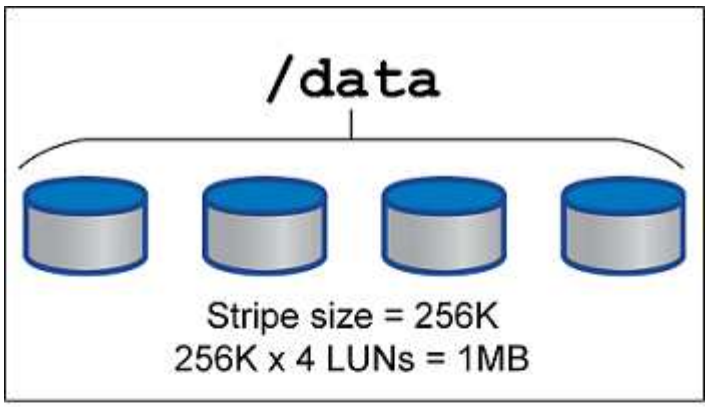
高いパフォーマンスを必要とするアプリケーションI/Oは、ほとんどの場合（a）基本ブロックサイズの単位または（b）1メガバイトのいずれかです。

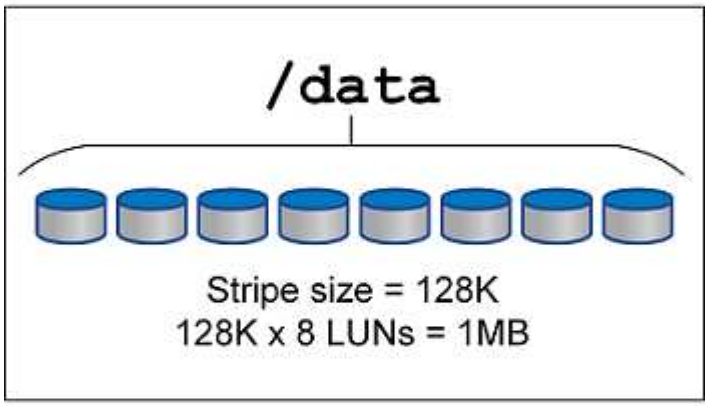
ストライピング構成の主な目的は、シングルファイルI/Oを1つのユニットとして実行し、マルチブロックI/O（サイズは1MB）をストライピングされたボリューム内のすべてのLUNで均等に並列化できるようにすることです。つまり、ストライプ・サイズはデータベース・ブロック・サイズより小さくすることはできず、ストライプ・サイズにLUN数を掛けたサイズは1MBにする必要があります。

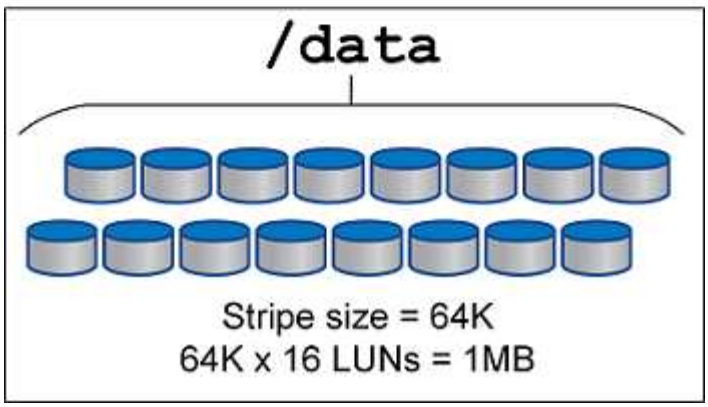
Oracle データベースでの LVM ストライピングのベスト プラクティス:

- ストライプ サイズ ≥ データベース ブロック サイズ。
- 最適な並列処理を実現するには、ストライプ サイズ * LUN の数 ≈ 1 MB になります。
- スループットを最大化し、ホットスポットを回避するには、ASMディスク グループごとに複数の LUN を使用します。

次の図に、ストライプサイズと幅の調整に使用できる3つのオプションを示します。LUNの数は、前述のパフォーマンス要件を満たすように選択されますが、いずれの場合も、1つのストライプ内の総データ量は1MBです。







NVFail

NVFAIL は、壊滅的なフェイルオーバー シナリオ中にデータの整合性を保証するONTAP機能です。

ASA r2 は簡素化された SAN アーキテクチャ (ボリュームの代わりに SAZ とストレージ ユニット) を使用しますが、この機能はASA r2 システムにも適用できます。

データベースは大きな内部キャッシュを保持しているため、ストレージ フェイルオーバー イベント中に破損する危険性が高くなります。壊滅的なイベントによりONTAPフェイルオーバーを強制する必要がある場合、全体的な構成の健全性に関係なく、結果として以前に確認された変更は事実上破棄される可能性があります。ストレージ アレイの内容は過去に遡り、データベース キャッシュの状態はディスク上のデータの状態を反映しなくなります。この不整合により、データ破損。

キャッシュはアプリケーション層またはサーバー層で発生する可能性があります。たとえば、プライマリ サイトとリモート サイトの両方でサーバーがアクティブになっている Oracle Real Application Cluster (RAC) 構成では、Oracle SGA 内にデータがキャッシュされます。強制フェイルオーバー操作によってデータが失われると、SGA に格納されているブロックがディスク上のブロックと一致しない可能性があるため、データベースが破損するリスクが生じます。

キャッシュのあまり知られていない使用法は、OS ファイル システム層です。プライマリ サイトにある LUN に基づくクラスター化されたファイル システムをリモート サイトのサーバーにマウントし、再度データをキャッシュすることができます。このような状況でNVRAMに障害が発生したり、強制的に引き継がれたりすると、ファイル システムが破損する可能性があります。

ONTAP は、NVFAIL とその関連設定を使用してデータベースとオペレーティング システムをこのシナリオから保護します。NVFAIL と関連設定は、キャッシュされたデータを無効にして、フェイルオーバー後に影響を受けるファイル システムを再マウントようにホストに信号を送ります。このメカニズムは、 AFF/ FASの場合と同様に、 ASA r2 LUN および名前空間に適用されます。

ASA r2 の主な考慮事項:

- NVFAIL は SAZ レベルではなく、LUN レベル (ストレージ ユニット) で動作します。
- Oracle データベースの場合、重要なコンポーネント (データファイル、REDO ログ、制御ファイル) をホストするすべての LUN で NVFAIL を有効にする必要があります。
- MetroClusterはASA r2 ではサポートされていないため、NVFAIL は主にローカル HA フェイルオーバー シナリオに適用されます。
- NFS はASA r2 ではサポートされていないため、NVFAIL の考慮事項は SAN ベースのワークロード (FC/iSCSI/NVMe) にのみ適用されます。

ASM 再利用ユーティリティ (ASRU)

ONTAP on ASA r2 は、インライン圧縮が有効になっている場合、LUN (ストレージ ユニット) に書き込まれたゼロ化されたブロックを効率的に削除します。Oracle ASM Reclamation Utility (ASRU) などのユーティリティは、未使用の ASM エクステントにゼロを書き込むことによって機能します。

これにより、DBA はデータが削除された後にストレージ アレイ上のスペースを再利用できるようになります。ONTAP はゼロをインターセプトし、LUN からスペースの割り当てを解除します。ストレージ システム内に実際のデータが書き込まれていないため、再利用プロセスは非常に高速です。

データベースに関しては、ASMディスクグループには0が含まれているため、LUNのこれらの領域を読み取ると0のストリームが生成されますが、ONTAPはドライブに0を格納しません。代わりに、メタデータが単純に変更され、LUNの初期化された領域がデータの空として内部的にマークされます。

同様の理由から、ゼロのブロックは実際にはストレージアレイ内で書き込みとして処理されないため、初期化されたデータを使用したパフォーマンステストは無効です。

ASA r2 ONTAPにおける ASRU の主な考慮事項:

- ASA r2 はブロックのみであるため、SAN ワークロードのAFF/ FASと同じように動作します。
- SAZ 内でプロビジョニングされた LUN および NVMe 名前空間に環境。
- FlexVolボリュームは存在しませんが、ゼロブロックの再利用動作は同じです。

ASRUを使用する場合は、Oracleが推奨するすべてのパッチがインストールされていることを確認してください。

仮想化

VMware、Oracle OLVM、KVMを使用したデータベースの仮想化は、最もミッションクリティカルなデータベースでさえ仮想化を選択したNetAppのお客様にとって、ますます一般的な選択肢となっています。

サポート性

Oracleによる仮想化のサポートポリシーについては、多くの誤解があります。特にVMware製品については、誤解が生じています。Oracle OUTRIGHTが仮想化をサポートしていないという話は珍しくありません。この概念は正しくないため、仮想化によるメリットを得る機会を逃してしまいます。実際の要件についてはOracleのドキュメントID 249212.1で説明されており、お客様が懸念事項と考えることはほとんどありません。

仮想化されたサーバで問題が発生し、これまでOracleサポートがその問題を認識していなかった場合は、物理ハードウェアで問題を再現するようにお客様に依頼されることがあります。最新バージョンの製品を使用しているOracleのお客様は、サポート性の問題が発生する可能性があるため、仮想化の使用を望まないかもしれませんが、一般提供されているOracle製品バージョンを使用しているお客様にとって、このような状況は現実のものではありません。

ストレージ提供

データベースの仮想化を検討しているお客様は、ビジネスニーズに基づいてストレージに関する意思決定を行う必要があります。これはすべてのIT意思決定に一般的に当てはまりますが、要件のサイズと範囲が大幅に異なるため、データベースプロジェクトでは特に重要です。

ストレージプレゼンテーションには、次の3つの基本的なオプションがあります。

- ハイパーバイザーデータストア上の仮想LUN
- iSCSI LUNをハイパーバイザーではなくVMのiSCSIイニシエータで管理
- （NFSベースのデータストアからではなく）VMによってマウントされたNFSファイルシステム
- 直接デバイスマッピング。VMware RDMはお客様から嫌われていますが、多くの場合、物理デバイスはKVMやOLVM仮想化と同様に直接マッピングされています。

パフォーマンス

仮想ゲストにストレージを提供する方法は、通常、パフォーマンスに影響しません。ホストOS、仮想ネットワークドライバ、ハイパーバイザーデータストアの実装はいずれも高度に最適化されており、基本的なベストプラクティスに従うかぎり、ハイパーバイザーとストレージシステムの間で使用可能なFCまたはIPネットワーク帯域幅をすべて消費できます。場合によっては、あるストレージプレゼンテーションのアプローチを使用した方が、別のアプローチよりもわずかに簡単に最適なパフォーマンスを得ることができますが、最終的な結果は同等である必要があります。

管理性

仮想ゲストにストレージをどのように提供するかを決定する際の重要な要素は、管理性です。正しい方法も間違った方法もありません。最適なアプローチは、IT運用のニーズ、スキル、好みによって異なります。

考慮すべき要素は次のとおりです。

- 透過性。 VMがファイルシステムを管理する場合、データベース管理者やシステム管理者は、データのファイルシステムのソースを簡単に特定できます。ファイルシステムとLUNへのアクセス方法は、物理サーバと同じです。
- 一貫性。 VMが自身のファイルシステムを所有している場合、ハイパーバイザーレイヤを使用するかどうかは管理性に影響します。プロビジョニング、監視、データ保護などの手順は、仮想環境と非仮想環境の両方を含め、資産全体で同じです。

  一方、完全に仮想化されたデータセンターでは、前述のように、プロビジョニング、保護、モンターリング、データ保護に同じ手順を使用できる一貫性という同じ根拠に基づいて、フットプリント全体でデータストアベースのストレージを使用することを推奨します。

- 安定性とトラブルシューティング。 VMが自身のファイルシステムを所有している場合、ストレージスタック全体がVM上に存在するため、優れた安定したパフォーマンスが提供され、問題のトラブルシューティングが簡単になります。ハイパーバイザーの役割は、FCフレームまたはIPフレームを転送することだけです。構成にデータストアが含まれていると、タイムアウト、パラメータ、ログファイル、および潜在的なバグが新たに発生するため、構成が複雑になります。
- モビリティ。 VMが自身のファイルシステムを所有している場合、Oracle環境を移動するプロセスははるかにシンプルになります。ファイルシステムは、仮想ゲストと非仮想ゲストの間で簡単に移動できます。
- *ベンダーロックイン。*データをデータストアに配置すると、別のハイパーバイザーを使用したり、仮想環境からデータを取り出すことが完全に困難になります。
- *スナップショットの有効化。*仮想環境での従来のバックアップ手順は、帯域幅が比較的限られているため、問題になる可能性があります。たとえば、多くの仮想データベースで日常的に必要とされるパフォーマンスには4ポート10GbEトランクで十分ですが、RMANなどのバックアップ製品を使用してバックアップを実行するには、データのフルサイズのコピーをストリーミングする必要がある場合は、このようなトランクでは不十分です。その結果、統合が進む仮想環境では、ストレージスナップショットを使用してバックアップを実行する必要が生じています。これにより、バックアップウィンドウで帯域幅とCPUの要件をサポートするためだけにハイパーバイザー構成を過剰に構築する必要がなくなります。

  ゲスト所有のファイルシステムを使用すると、保護を必要とするストレージオブジェクトをより簡単にターゲットにできるため、Snapshotベースのバックアップとリストアを簡単に活用できる場合があります。しかし、データストアやスナップショットとうまく統合できる仮想化データ保護製品の数はますます増えています。仮想化されたホストにストレージを提供する方法を決定する前に、バックアップ戦略を十分に検討する必要があります。

準仮想化ドライバ

最適なパフォーマンスを実現するには、準仮想化ネットワークドライバを使用することが重要です。データストアを使用する場合は、準仮想化SCSIドライバが必要です。準仮想化デバイスドライバを使用すると、エミュレートされたドライバとは対照的に、ゲストをハイパーバイザーにより深く統合できます。エミュレートされたドライバでは、ハイパーバイザーは物理ハードウェアの動作を模倣するためにより多くのCPU時間を消費します。

RAMのオーバーコミット

RAMのオーバーコミットとは、物理ハードウェア上に存在するよりも多くの仮想RAMをさまざまなホストに設定することを意味します。原因で予期しないパフォーマンスの問題が発生する可能性があります。データベースを仮想化する場合、Oracle SGAの基盤となるブロックがハイパーバイザーによってストレージにスワップアウトされないようにする必要があります。このように設定すると、パフォーマンスが非常に不安定になります。

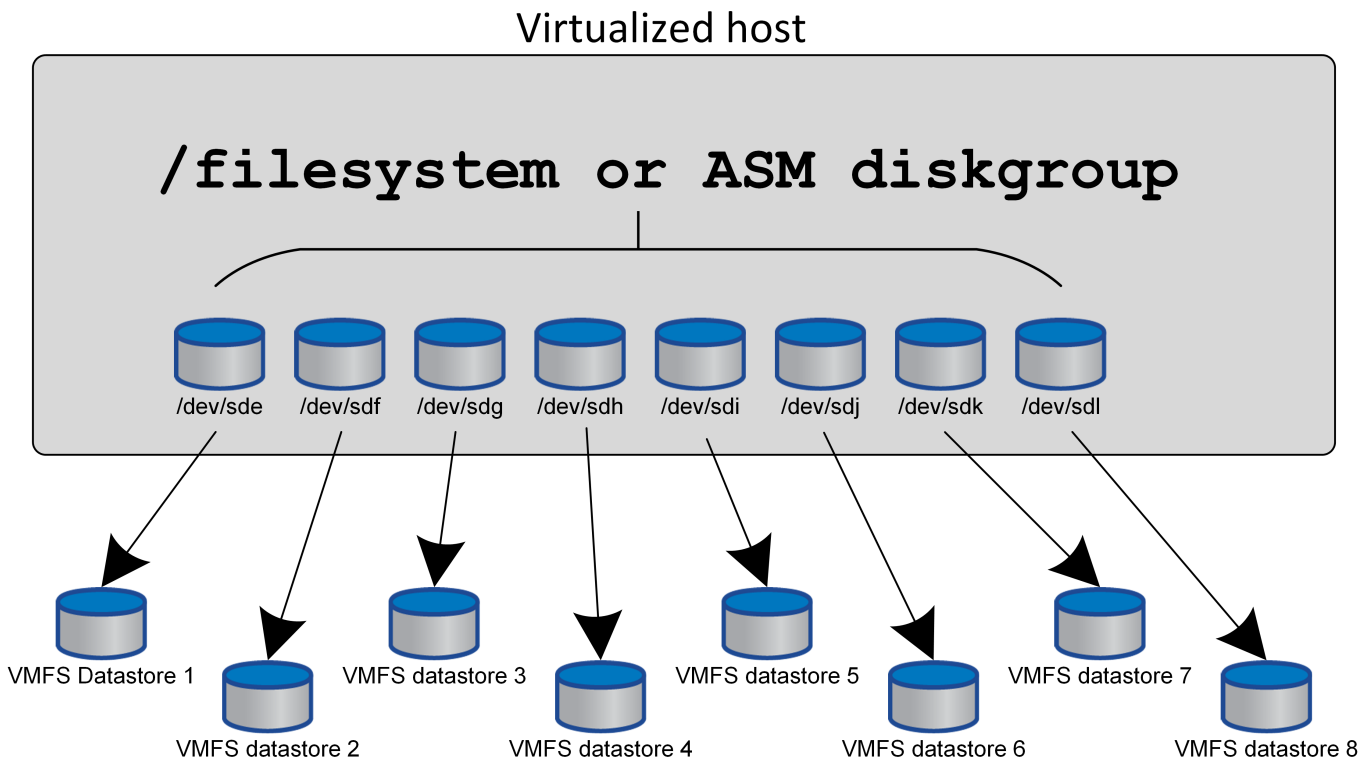
データストアのストライピング

データストアでデータベースを使用する場合、パフォーマンスストライピングに関して考慮すべき重要な要素が1つあります。

VMFSなどのデータストアテクノロジは複数のLUNにまたがることができますが、ストライプデバイスではありません。LUNが連結されます。その結果、LUNのホットスポットが発生する可能性があります。たとえば、一般的なOracleデータベースに8 LUNのASMディスクグループがあるとします。8つの仮想化されたLUNはすべて8 LUNのVMFSデータストアにプロビジョニングできますが、データがどのLUNに格納されるかは保証されません。この構成では、8つの仮想LUNすべてがVMFSデータストア内の1つのLUNを占有するようになります。これがパフォーマンスのボトルネックになります。

通常、ストライピングが必要です。KVMなどの一部のハイパーバイザーでは、次の説明に従ってLVMストライピングを使用してデータストアを構築できます。 "こちらをご覧ください"。VMwareの場合、アーキテクチャは少し異なります。各仮想LUNを別 々 のVMFSデータストアに配置する必要があります。

例：



このアプローチの主な推進力はONTAPではありません。これは、1つのVMまたはハイパーバイザーLUNが並行して処理できる処理数に固有の制限があるためです。1つのONTAP LUNでサポートできるIOPSは、通常、ホストが要求できるIOPSよりもはるかに多くなります。単一LUNのパフォーマンス制限は、ほとんどの場合、ホストOSが原因です。そのため、ほとんどのデータベースでは、パフォーマンスのニーズを満たすために4~8個のLUNが必要になります。

VMwareアーキテクチャでは、データストアやLUNパスの最大数がこのアプローチで発生しないように、アーキテクチャを慎重に計画する必要があります。また、すべてのデータベースに固有のVMFSデータストアセットを用意する必要はありません。主に必要なのは、各ホストに、仮想化されたLUNからストレージシステム自体のバックエンドLUNへの4 ~ 8個のIOパスのクリーンなセットがあることを確認することです。まれに、より多くのデータストアが本当に極端なパフォーマンス要求に対して有益な場合もありますが、一般に、全データベースの95%に対して4~8個のLUNで十分です。8個のLUNを含む1つのONTAPボリュームでは、一般的なOS / ONTAP /ネットワーク構成で、Oracleブロックのランダム IOPSを最大25、000個サポートできます。

階層化

概要

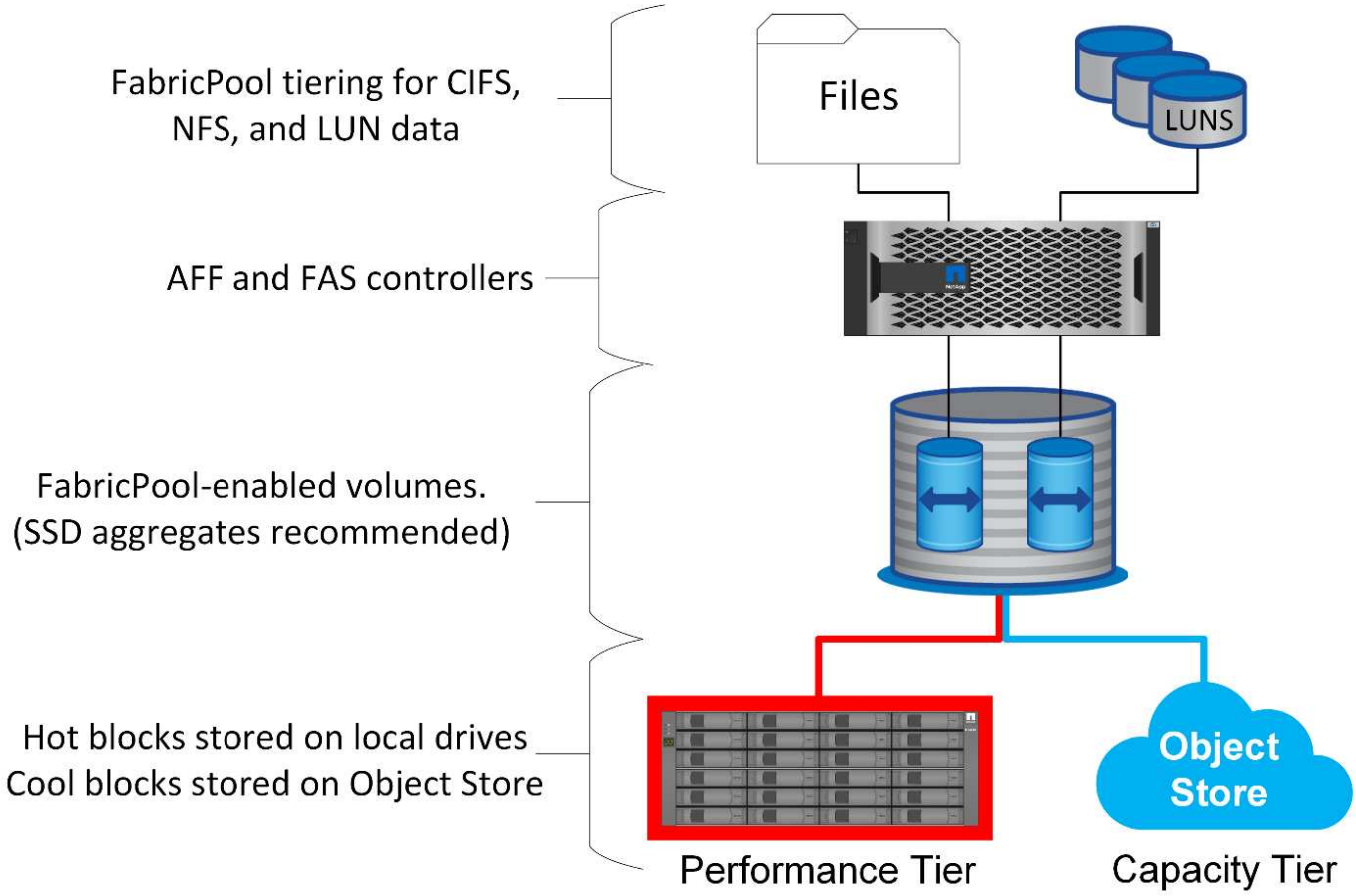
FabricPoolの階層化がOracleやその他のデータベースに与える影響を理解するには、低レベルのFabricPoolアーキテクチャについて理解しておく必要があります。

アーキテクチャ

FabricPoolは、ブロックをホットまたはクールに分類し、最も適切なストレージ階層に配置する階層化テクノロジです。ほとんどの場合、パフォーマンス階層はSSDストレージに配置され、ホットデータブロックをホストします。大容量階層はオブジェクトストアに配置され、クールデータブロックをホストします。サポートされるオブジェクトストレージには、NetApp StorageGRID、ONTAP S3、Microsoft Azure Blobストレージ、Alibaba Cloud Object Storageサービス、IBM Cloud Object Storage、Google Cloudストレージ、Amazon

AWS S3などがあります。

ブロックをホットまたはクールに分類する方法を制御する複数の階層化ポリシーを使用できます。ポリシーはボリューム単位で設定し、必要に応じて変更できます。パフォーマンス階層と大容量階層の間で移動されるのはデータブロックのみです。LUNとファイルシステムの構造を定義するメタデータは、常にパフォーマンス階層に残ります。そのため、管理はONTAPに一元化されます。ファイルとLUNは、他のONTAP構成に格納されているデータと変わりません。NetApp AFFコントローラまたはFASコントローラが定義されたポリシーを適用して、データを適切な階層に移動します。



オブジェクトストレージプロバイダ

オブジェクトストレージプロトコルでは、単純なHTTP要求またはHTTPS要求を使用して大量のデータオブジェクトを格納します。ONTAPからのデータアクセスは要求の迅速な処理に依存するため、オブジェクトストレージへのアクセスは信頼できるものでなければなりません。オプションには、Amazon S3の[Standard and InFrequent Access]オプション、Microsoft Azure Hot and Cool Blob Storage、IBM Cloud、Google Cloudなどがあります。Amazon GlacierやAmazon Archiveなどのアーカイブオプションはサポートされていません。データの読み出しに要する時間がホストのオペレーティングシステムやアプリケーションの許容範囲を超える可能性があるためです。

NetApp StorageGRIDもサポートされており、最適なエンタープライズクラスの解決策です。ハイパフォーマンス、拡張性、セキュリティに優れたオブジェクトストレージシステムであり、FabricPoolデータだけでなく、エンタープライズアプリケーション環境に組み込まれる可能性が高まるその他のオブジェクトストレージアプリケーションにも、地理的な冗長性を提供します。

また、StorageGRIDは、多くのパブリッククラウドプロバイダがサービスからデータを読み返す際に出力料金を課す必要がないため、コストを削減できます。

データとメタデータ

ここで「data」という用語は、メタデータではなく実際のデータブロックを環境することに注意してください。データブロックのみが階層化され、メタデータはパフォーマンス階層に残ります。また、あるブロックのステータス（hotまたはcool）が影響を受けるのは、実際のデータブロックを読み取った場合のみです。ファイルの名前、タイムスタンプ、所有権のメタデータを読み取っても、基盤となるデータブロックの場所には影響しません。

バックアップ

FabricPoolはストレージの設置面積を大幅に削減できますが、それだけではバックアップ解決策ではありません。NetApp WAFLメタデータは常に高パフォーマンス階層に残ります。大容量階層にはWAFLメタデータが含まれていないため、大容量階層のデータを使用して新しい環境を作成することはできません。

ただし、FabricPoolはバックアップ戦略の一部になる可能性があります。たとえば、FabricPoolにはNetApp SnapMirrorレプリケーションテクノロジを設定できます。ミラーの各半分は、オブジェクトストレージターゲットに独自に接続できます。その結果、データの2つの独立したコピーが作成されます。プライマリコピーは、パフォーマンス階層のブロックと大容量階層内の関連するブロックで構成され、レプリカはパフォーマンスブロックと容量ブロックの2つ目のセットです。

階層化ポリシー

階層化ポリシー

ONTAPでは4つのポリシーを使用して、高パフォーマンス階層にあるOracleデータを大容量階層に再配置する方法を制御できます。

Snapshotのみ

。 `snapshot-only` `tiering-policy` アクティブファイルシステムと共有されていないブロックにのみ適用されます。基本的には、データベースバックアップの階層化につながります。Snapshotが作成されてそのブロックが上書きされると、ブロックが階層化の候補となり、その結果、Snapshot内にのみ存在するブロックが作成されます。Aの前の遅延 `snapshot-only` ブロックは冷却されていると見なされ、によって制御されます。 `tiering-minimum-cooling-days` ボリュームの設定。ONTAP 9.8で指定できる範囲は2～183日です。

多くのデータセットは変更率が低いため、このポリシーによる削減効果は最小限に抑えられます。たとえば、ONTAPで観察される一般的なデータベースの変更率は、週あたり5%未満です。データベースのアーカイブログは大量のスペースを占有することがありますが、通常はアクティブファイルシステムに引き続き存在するため、このポリシーでは階層化の対象になりません。

自動

。 `auto` 階層化ポリシーは、階層化をSnapshot固有のブロックだけでなく、アクティブなファイルシステム内のブロックにも拡張します。ブロックが冷却されるまでの遅延は、 `tiering-minimum-cooling-days` ボリュームの設定。ONTAP 9.8で指定できる範囲は2～183日です。

このアプローチでは、では使用できない階層化オプションが有効になります。 `snapshot-only` ポリシー：たとえば、データ保護ポリシーで特定のログファイルを90日間保持する必要がある場合があります。クーリング期間を3日に設定すると、3日を超過した古いログファイルがパフォーマンスレイヤから階層化されます。この操作により、パフォーマンス階層のかなりのスペースが解放されると同時に、90日間分のすべてのデータを表示して管理することができます。

なし

。 none 階層化ポリシーを使用すると、追加のブロックがストレージレイヤから階層化されなくなりますが、大容量階層のデータは読み取りが行われるまで大容量階層に残ります。その後ブロックが読み取られると、元に戻されてパフォーマンス階層に配置されます。

を使用する主な理由は、 none 階層化ポリシーはブロックが階層化されないようにするためのものですが、時間の経過とともにポリシーを変更すると便利です。たとえば、あるデータセットが大容量レイヤに階層化されているとしますが、完全なパフォーマンス機能が予期せず必要になったとします。このポリシーを変更すると、追加の階層化が不要になり、I/Oの増加に伴って読み取られたブロックがパフォーマンス階層に残るようにすることができます。

すべて

。 all 階層化ポリシーで置き換えられる backup ONTAP 9.6以降のポリシー。。 backup データ保護ボリューム（SnapMirrorまたはNetApp SnapVaultのデスティネーション）にのみ適用されるポリシー。。 all ポリシーの機能は同じですが、データ保護ボリュームに限定されません。

このポリシーでは、ブロックはすぐにクールとみなされ、すぐに容量レイヤに階層化できるようになります。

このポリシーは、長期的なバックアップに特に適しています。Hierarchical Storage Management（HSM；階層型ストレージ管理）の一種としても使用できます。以前は、ファイルシステム上でファイル自体を認識したまま、ファイルのデータブロックをテープに階層化するためにHSMが一般的に使用されていました。FabricPoolボリューム all ポリシーを使用すると、表示および管理可能なにファイルを格納できますが、ローカルストレージ階層のスペースはほとんど消費しません。

読み出しポリシー

階層化ポリシーは、どのOracleデータベースブロックをパフォーマンス階層から大容量階層に階層化するかを制御します。読み出しポリシーは、階層化されたブロックが読み取られたときの処理を制御します。

デフォルト

すべてのFabricPoolボリュームの初期設定は `default`これは、動作が「cloud-retrieval-policy」によって制御されることを意味します。'正確な動作は、使用する階層化ポリシーによって異なります。

- auto-ランダムリードデータのみを取得
- snapshot-only-すべてのシーケンシャルまたはランダムリードデータを取得
- none-すべてのシーケンシャルまたはランダムリードデータを取得
- all-大容量階層からデータを取得しない

オンリード

設定 cloud-retrieval-policy をオンリードに設定するとデフォルトの動作が無効になるため、階層化されたデータが読み取られた場合、そのデータはパフォーマンス階層に返されます。

たとえば、ボリュームは、 auto 階層化ポリシーとほとんどのブロックが階層化されます。

ビジネスニーズの予期しない変化によって、特定のレポートを作成するために一部のデータを繰り返しスキャンする必要がある場合は、 cloud-retrieval-policy 終了： on-read シーケンシャルデータとランダム

リードデータの両方を含む、読み取りされるすべてのデータがパフォーマンス階層に返されます。これにより、ボリュームに対するシーケンシャルI/Oのパフォーマンスが向上します。

プロモート

昇格ポリシーの動作は階層化ポリシーによって異なります。階層化ポリシーがの場合 `auto`をクリックし、`cloud-retrieval-policy `to `promote 次回の階層化スキャンで大容量階層のすべてのブロックを戻します。

階層化ポリシーがの場合 `snapshot-only`を指定すると、アクティブファイルシステムに関連付けられているブロックのみが返されます。通常、これは効果がありません。これは、 `snapshot-only ポリシーは、Snapshotにのみ関連付けられたブロックになります。アクティブファイルシステムに階層化されたブロックはありません。

ただし、ボリュームSnapRestoreまたはSnapshotからのファイルクローン操作によってボリューム上のデータがリストアされた場合、Snapshotにのみ関連付けられていたために階層化されたブロックの一部がアクティブファイルシステムで必要になることがあります。一時的に `cloud-retrieval-policy` ポリシーの宛先 `promote` ローカルで必要なすべてのブロックを迅速に取得できます。

なし

大容量階層からブロックを取得しないでください。

階層化戦略

完全なファイル階層化

FabricPool階層化はブロックレベルで動作しますが、ファイルレベルの階層化に使用できる場合もあります。

多くのアプリケーションデータセットは日付別に整理されており、そのようなデータが古くなるにつれてアクセスされる可能性はますます低くなっています。たとえば、銀行が5年間の顧客明細書を含むPDFファイルのリポジトリを持っていても、最近の数か月のみがアクティブになっているとします。FabricPoolを使用して、古いデータファイルを大容量階層に再配置できます。クーリング期間を14日間にすると、直近の14日間のPDFファイルがパフォーマンス階層に残ります。さらに、少なくとも14日ごとに読み取られたファイルはホットのままであるため、パフォーマンス階層に残ります。

ポリシー

ファイルベースの階層化アプローチを実装するには、ファイルが書き込まれ、その後変更されないようにする必要があります。。 `tiering-minimum-cooling-days` 必要なファイルが高パフォーマンス階層に残るように、ポリシーを十分に高く設定する必要があります。たとえば、最新の60日分のデータが必要なデータセットで、最適なパフォーマンス保証が設定されているとします。 `tiering-minimum-cooling-days` 60までの期間。ファイルアクセスパターンに基づいても同様の結果が得られます。たとえば、最新の90日間のデータが必要で、アプリケーションがその90日間のデータにアクセスしている場合、データは高パフォーマンス階層に残ります。を設定する `tiering-minimum-cooling-days` 2までの期間では、データの使用頻度が低下した後、迅速な階層化が行われます。

。 `auto` これらのブロックの階層化を推進するにはポリシーが必要です。これは、 `auto` ポリシーは、アクティブファイルシステム内のブロックに影響します。

データにアクセスすると、ヒートマップデータがリセットされます。ウィルススキャン、インデックス作成、さらにはソースファイルを読み取るバックアップ処理も行われるため、 `tiering-minimum-cooling-days` しきい値に達していません。

部分的なファイル階層化

FabricPoolはブロックレベルで機能するため、変更の可能性があるファイルは部分的にオブジェクトストレージに階層化し、部分的にパフォーマンス階層に残すことができます。

これはデータベースで一般的です。アクセス頻度の低いブロックが含まれていることがわかっているデータベースも、FabricPool階層化の候補になります。たとえば、サプライチェーン管理データベースには履歴情報が含まれている可能性があります。この情報は、必要に応じて利用できなければなりませんが、通常の運用中はアクセスできません。FabricPoolを使用すると、非アクティブなブロックを選択的に再配置できます。

たとえば、FabricPoolで実行されているデータファイルの場合、 `tiering-minimum-cooling-days` 90日の期間には、過去90日間にアクセスされたブロックがパフォーマンス階層に保持されます。ただし、90日間アクセスされなかったデータはすべて大容量階層に再配置されます。それ以外の場合は、通常のアプリケーションアクティビティで正しいブロックが正しい階層に保持されます。たとえば、データベースが通常、過去60日間のデータを定期的に処理するために使用されている場合、 `tiering-minimum-cooling-days` 期間を設定できるのは、アプリケーションの自然なアクティビティによって、ブロックが早期に再配置されないようにするためです。

。 `auto` データベースには注意してポリシーを使用する必要があります。多くのデータベースには、四半期末のプロセスやインデックスの再作成などの定期的なアクティビティがあります。これらの操作の期間が `tiering-minimum-cooling-days` パフォーマンスに問題が生じる可能性があります。たとえば、四半期末の処理で1TBのデータをまだ使用していない状態で処理する必要がある場合、そのデータは大容量階層に配置される可能性があります。大容量階層からの読み取りは非常に高速であることが多く、原因のパフォーマンスに問題はない可能性がありますが、正確な結果はオブジェクトストアの設定によって異なります。

ポリシー

。 `tiering-minimum-cooling-days` パフォーマンス階層で必要になる可能性のあるファイルを保持できるように、ポリシーを十分な高さに設定する必要があります。たとえば、最新の60日分のデータが必要でパフォーマンスが最適なデータベースでは、 `tiering-minimum-cooling-days` 60日までの期間。同様の結果は、ファイルのアクセスパターンに基づいても達成できます。たとえば、最新の90日間のデータが必要で、アプリケーションがその90日間のデータにアクセスしている場合、データは高パフォーマンス階層に残ります。を設定します `tiering-minimum-cooling-days` データの使用頻度が低下した場合は、2日間の期間でデータが階層化されます。

。 `auto` これらのブロックの階層化を推進するにはポリシーが必要です。これは、 `auto` ポリシーは、アクティブファイルシステム内のブロックに影響します。

データにアクセスすると、ヒートマップデータがリセットされます。そのため、データベースのテーブル全体がスキャンされ、ソースファイルを読み取るバックアップアクティビティも行われるため、 `tiering-minimum-cooling-days` しきい値に達していません。

アーカイブログの階層化

FabricPoolの最も重要な用途は、データベーストランザクションログなどの既知のコールドデータの効率化です。

ほとんどのリレーショナルデータベースは、ポイントインタイムリカバリを実現するためにトランザクションログアーカイブモードで動作します。データベースへの変更は、トランザクションログに変更を記録することによってコミットされ、トランザクションログは上書きされずに保持されます。そのため、大量のアーカイブトランザクションログを保持しなければならない場合があります。同様の例は、保持する必要があるデータを生成する他の多くのアプリケーションワークフローにも存在しますが、アクセスされることはほとんどありません。

FabricPoolは、階層化が統合された単一の解決策を提供することで、これらの問題を解決します。ファイルは通常の場所に保存されてアクセス可能な状態に維持されますが'プライマリ・アレイのスペースはほとんど消費されません

ポリシー

を使用します `tiering-minimum-cooling-days` 数日間のポリシーを設定すると、最近作成されたファイル（短期的に必要になる可能性が高いファイル）のブロックが高パフォーマンス階層に保持されます。その後、古いファイルのデータブロックが大容量階層に移動されます。

。 `auto` ログが削除されたか、プライマリファイルシステムに引き続き存在しているかに関係なく、クーリングしきい値に達したときに、プロンプト階層化を適用します。必要となる可能性があるすべてのログをアクティブファイルシステムの1つの場所に格納することも、管理を簡易化します。リストアが必要なファイルを特定するためにSnapshotを検索する必要はありません。

Microsoft SQL Serverなどの一部のアプリケーションでは、バックアップ処理中にトランザクションログファイルが切り捨てられ、ログがアクティブファイルシステムに記録されなくなります。容量は、 `snapshot-only` 階層化ポリシー `auto` アクティブファイルシステムにはログデータが冷却されることはほとんどないため、ログデータにはポリシーは役立ちません。

Snapshotの階層化

FabricPoolの初期リリースでは、バックアップのユースケースを対象としていました。階層化できるブロックのタイプは、アクティブファイルシステム内のデータに関連付けられなくなったブロックだけです。そのため、大容量階層に移動できるのはSnapshotデータブロックだけです。これは、パフォーマンスに影響を与えないようにする必要がある場合に、最も安全な階層化オプションの1つです。

ポリシー-ローカル**Snapshot**

アクセス頻度の低いSnapshotブロックを大容量階層に階層化する方法は2つあります。まず、 `snapshot-only` ポリシーはSnapshotブロックのみを対象としています。ただし、 `auto` ポリシーには、 `snapshot-only` ブロックの場合は、アクティブファイルシステムのブロックも階層化されます。これは望ましくない可能性があります。

。 `tiering-minimum-cooling-days` この値は、リストア時に必要となる可能性のあるデータを高パフォーマンス階層で使用できるようにする期間に設定する必要があります。たとえば、重要な本番環境データベースのリストアシナリオのほとんどには、過去数日間のある時点のリストアポイントが含まれます。セッテイ `tiering-minimum-cooling-days` 値を3に設定すると、ファイルをリストアしたときにパフォーマンスがすぐに最大になるようにファイルが作成されます。アクティブファイル内のすべてのブロックは、大容量階層

からリカバリすることなく高速ストレージに残ります。

ポリシー-レプリケートされた**Snapshot**

リカバリのみに使用されるSnapMirrorまたはSnapVaultでレプリケートされるSnapshotには、一般にFabricPoolを使用する必要があります。 `all` ポリシー：このポリシーでは、メタデータはレプリケートされますが、すべてのデータブロックがただちに大容量階層に送信されるため、パフォーマンスが最大限に向上します。ほとんどのリカバリプロセスではシーケンシャルI/Oが発生しますが、これは本質的に効率的です。オブジェクトストアのデスティネーションからのリカバリ時間を評価する必要がありますが、適切に設計されたアーキテクチャでは、このリカバリプロセスにローカルデータからのリカバリよりも大幅に時間がかかる必要はありません。

レプリケートされたデータをクローニングにも使用する場合は、 `auto` ポリシーはより適切であり、 `tiering-minimum-cooling-days` クローニング環境で定期的に使用されることが期待されるデータを含む価値。たとえば、データベースのアクティブなワーキングセットには、過去3日間に読み書きされたデータが含まれている場合がありますが、さらに6カ月分の履歴データが含まれている場合もあります。その場合は、 `auto` SnapMirrorデスティネーションでポリシーを設定すると、作業セットを高パフォーマンス階層で使用できるようになります。

バックアップの階層化

従来のアプリケーションバックアップには、Oracle Recovery Managerなどの製品が含まれています。Oracle Recovery Managerは、元のデータベースの場所以外にファイルベースのバックアップを作成します。

```
 `tiering-minimum-cooling-days` policy of a few days preserves the most
recent backups, and therefore the backups most likely to be required for
an urgent recovery situation, on the performance tier. The data blocks of
the older files are then moved to the capacity tier.
```

。 `auto`
ポリシーは、バックアップデータに最も適したポリシーです。これにより、ファイルが削除されたか、プライマリファイルシステムに引き続き存在しているかに関係なく、クーリングしきい値に達したときに迅速に階層化されます。必要となる可能性があるすべてのファイルをアクティブファイルシステムの1つの場所に格納することも、管理を簡易化します。リストアが必要なファイルを特定するために`Snapshot`を検索する必要はありません。

。 `snapshot-only` ポリシーは機能するように設定できますが、アクティブファイルシステムに存在しなくなった環境ブロックのみが対象となります。そのため、データを階層化するには、まずNFS共有またはSMB共有上のファイルを削除する必要があります。

LUNからファイルを削除するとファイル参照がファイルシステムのメタデータから削除されるだけなので、このポリシーはLUN設定の場合はさらに効率的ではありません。LUN上の実際のブロックは、上書きされるまでそのまま維持されます。このような状況では、ファイルが削除されてブロックが上書きされて階層化の候補になるまでに長時間の遅延が発生する可能性があります。の移動にはいくつかの利点があります。 `snapshot-only` ブロックは大容量階層に移動しますが、全体的にはバックアップデータのFabricPool管理が最適なのは、 `auto` ポリシー：

このアプローチは、バックアップに必要なスペースをより効率的に管理するのに役立ちますが、FabricPool自体はバックアップテクノロジではありません。バックアップファイルをオブジェクトストアに階層化すると、ファイルは元のストレージシステムに引き続き表示されるため、管理が簡易化されますが、オブジェクトストアデスティネーションのデータブロックは元のストレージシステムに依存します。ソースボリュームが失われると、オブジェクトストアのデータを使用できなくなります。

オブジェクトストアへのアクセスの中断

FabricPoolでデータセットを階層化すると、プライマリストレージアレイとオブジェクトストア階層の間に依存関係が生じます。オブジェクトストレージには、さまざまなレベルの可用性を提供するオプションが多数あります。プライマリストレージアレイとオブジェクトストレージ階層の間の接続が失われた場合の影響を理解することが重要です。

ONTAPに対して実行するI/Oで大容量階層のデータが必要になり、ONTAPが大容量階層に到達してブロックを読み出すことができない場合、最終的にI/Oはタイムアウトします。このタイムアウトの影響は、使用するプロトコルによって異なります。NFS環境では、ONTAPはプロトコルに応じてEJUKEBOXまたはEDELAYのいずれかの応答で応答します。一部の古いオペレーティングシステムではエラーと解釈される場合がありますが、現在のオペレーティングシステムとOracle Direct NFSクライアントの現在のパッチレベルでは、これを再試行可能なエラーとして扱い、I/Oの完了を待ち続けます。

環境SAN環境のタイムアウトを短縮します。オブジェクトストア環境のブロックが必要で、2分間アクセスできない場合は、読み取りエラーがホストに返されます。ONTAPボリュームとLUNはオンラインのままですが、ホストOSからファイルシステムにエラー状態のフラグが設定されることがあります。

オブジェクトストレージの接続の問題 `snapshot-only` バックアップデータのみが階層化されるため、ポリシーはそれほど重要ではありません。通信に問題があると、データのリカバリに時間がかかりますが、それ以外の場合はアクティブに使用されているデータに影響。 `auto` および `all` ポリシーを使用すると、アクティブなLUNからコールドデータを階層化できます。つまり、オブジェクトストアデータの読み出し中にエラーが発生すると、データベースの可用性に影響する可能性があります。これらのポリシーを使用したSAN環境は、高可用性を実現するように設計されたエンタープライズクラスのオブジェクトストレージとネットワーク接続でのみ使用してください。NetApp StorageGRIDは優れたオプションです。

Oracleのデータ保護

ONTAPによるデータ保護

NetAppは、最もミッションクリティカルなデータがデータベースに含まれていることを認識しています。

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

このドキュメントは、以前に公開されていたテクニカルレポート_TR-4591：『Oracle data protection：Backup、recovery、and replication』に代わるものです。_

計画

適切なエンタープライズデータ保護アーキテクチャは、さまざまなイベントにおけるデータの保持、リカバリ性、耐障害性に関するビジネス要件に依存します。

たとえば、対象となるアプリケーション、データベース、重要なデータセット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、一般的なSLAへの準拠を保証する単一データセットのバックアップ戦略の構築は非常に簡単です。データセットの数が増えるにつれて監視が複雑になり、バックアップの失敗に対処するために、管理者がますます多くの時間を費やすことになる可能性があります。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。

データセットのサイズも戦略に影響します。たとえば、データセットが非常に小さいため、100GBのデータベースのバックアップとリカバリには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。通常、100TBのデータベースでは、RTOによって複数日の停止が許容される場合を除き、まったく異なる戦略が必要になります。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さまざまな要因があります。たとえば、重要な本番環境のアクティビティをサポートしているデータベースがあり、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。

RTO、RPO、SLA計画

ONTAPを使用すると、Oracleデータベースのデータ保護戦略をビジネス要件に簡単にカスタマイズできます。

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、さまざまなデータリカバリシナリオを検討する必要があります。たとえば、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から、サイト全体の損失を含むディザスタリカバリのシナリオまで、さまざまなシナリオを検討する必要があります。

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。また、最初に見落とされた要件は、プロジェクトを間違った方向に進めたり、直前の設計変更を必要としたりす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ことは非常に不便ですが、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。RTOをゼロにすることはできません。これは、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であるためです。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。多くの場合、RPOはSnapshotまたはSnapMirror更新の頻度によって決まります。

場合によっては、RPOをより積極的に設定し、特定のデータをより頻繁に選択的に保護することができます。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、つまりデータ損失がないはずです。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のすべてのデータが削除される可能性があります。唯一の方法は、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あり、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。しかし、データが絶えず移動し、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であるため、これには限界があります。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。これには、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。一部のアプリケーション環境では、比較的通常のユーザエラーからデータセンターの破壊に至るまで、データ損失の状況に対して、RPOゼロとRTOほぼゼロを達成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の要件は、次の2つの理由で厳しいものにする必要があります。まず、データ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、ほぼ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。容易に修復できる重大なリスクに対処しない理由はありません。そのため、ローカルリカバリのRPOとRTOの目標を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、災害が発生する可能性や、関連するデータの損失やビジネスの中断がもたらす影響によって大きく異なります。RPOとRTOの要件は、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いる必要があります。論理的および物理的な複数の災害シナリオを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破損、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を特に標的にされる企業は、ますます増えています。この脅威に対して、NetAppには改ざん防止スナップショットが用意されており、ストレージ管理者であっても、設定された有効期限までに保護されたデータを変更することはできません。

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保護ではディスクドライブの冗長性が提供され、HBAを使用することで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。

エンタープライズ環境では、通常、サイト全体のインフラストラクチャ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がサイトの完全な損失である時点まで保護することができます。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、アプリケーションやデータベースの処理を続行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が許容できない場合があり、同期ミラーリングの使用が除外されます。
- 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、パフォーマンスと柔軟性の向上を可能にする非同期ソリューションの両方を含むソリューションを提供しています。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製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。

データベースの可用性

ONTAPは、Oracleデータベースの可用性を最大限に高めるように設計されています。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への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（データファイル）上の永続的データの更新は、ONTAP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。通常運用時は、NVRAMデータもデータベースのREDOログも読み取られません。

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にある物理インターフェイス上の一時的な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。

> 大規模なクラスタ内のノード間でデータを再配置できるように、追加のコントローラへの追加のパスを設定することもできますが、これは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通常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処理を再送信するために、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。これにより、ネットワークスイッチはメディアアクセス制御（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の停止は、通常数秒で測定され、多くの場合は検出されません。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、再試行が必要な大量のI/Oが短時間にキューイングされる場合があります。これにより、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されるまでにさらに時間がかかる場合があります。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。

次の表に、ストレージシステムがアプリケーション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。どのアプリケーション環境にもエラーは発生しません。テイクオーバーはI/O処理の一時停止として表示されます。

	NFS	AFF	ASA
計画的テイクオーバー	15秒	6～10秒	2～3秒
計画外のテイクオーバー	30秒	6～10秒	2～3秒

チェックサムとデータ整合性

ONTAPとそのサポートされているプロトコルには、保存データとネットワーク経由で転送されるデータの両方を含む、Oracleデータベースの整合性を保護する複数の機能が含まれています。

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。

- データを破損から保護する必要があります。
- データはドライブ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。後続の読み取りでブロックが識別されていても、メタデータにブロックが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。説明は非常に複雑ですが、基本的にONTAP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。これは、ドライブによる書き込みが完了しなかったことを示します。

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。これは、ONTAPでパリティドライブを使用して行われます。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。これにより、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くのI/Oが追加で作成されます。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を継続して使用することでデータ損失のリスクが増加しなかったとしても、データ損失の影響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多いほど、データのリカバリにかかる時間が長くなり、ビジネス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技術を開発した。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。ドライブのサイズは拡大を続けており、その結果、NetAppは3つ目のパリティドライブを導入するNetApp RAID-TECテクノロジを開発しました。

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。2本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劣ります。

また、パフォーマンス上の懸念から、RAID-4 / 5 / 6よりもRAID-10が推奨されることを示す履歴データベースのベストプラクティスもいくつかあります。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一般的に正しいものですが、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。つまり、書き込み処理は少なくとも2つの場所に安全に格納する必要があります。

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。

ハードウェア障害からの保護：**NVFAIL**

前述したように、書き込みの確認応答は、少なくとも1台の他のコントローラでローカルのNVRAMとNVRAMに記録されるまで返されません。このアプローチにより、ハードウェア障害や停電が発生しても、転送中のI/Oが失われることはありません。ローカルのNVRAMに障害が発生したり、HAパートナーへの接続に障害が発生したりすると、この実行中のデータはミラーリングされなくなります。

ローカルNVRAMからエラーが報告されると、ノードはシャットダウンします。このシャットダウンにより、HAパートナーコントローラにフェイルオーバーします。障害が発生したコントローラが書き込み処理を確認していないため、データが失われることはありません。

データが同期されていない場合、ONTAPは、強制的にフェイルオーバーを実行しない限り、フェイルオーバーを許可しません。この方法で条件を変更すると、元のコントローラにデータが残っている可能性があり、データ損失が許容されることが確認されます。

データベースはディスク上のデータの大規模な内部キャッシュを保持しているため、フェイルオーバーが強制された場合、データベースが破損する可能性が特に高くなります。強制的なフェイルオーバーが発生した場合、以前に承認された変更は事実上破棄されます。ストレージアレイの内容は実質的に時間を逆方向に移動し、データベースキャッシュの状態はディスク上のデータの状態を反映しなくなります。

この状況からデータを保護するために、ONTAPでは、NVRAMの障害に対する特別な保護をボリュームに設定できます。この保護メカニズムがトリガーされると、ボリュームがNVFAILという状態になります。この状態では、古いデータを使用しないように原因Aアプリケーションをシャットダウンするl/Oエラーが発生します。確認済みの書き込みがストレージアレイに存在する必要があるため、データは失われません。

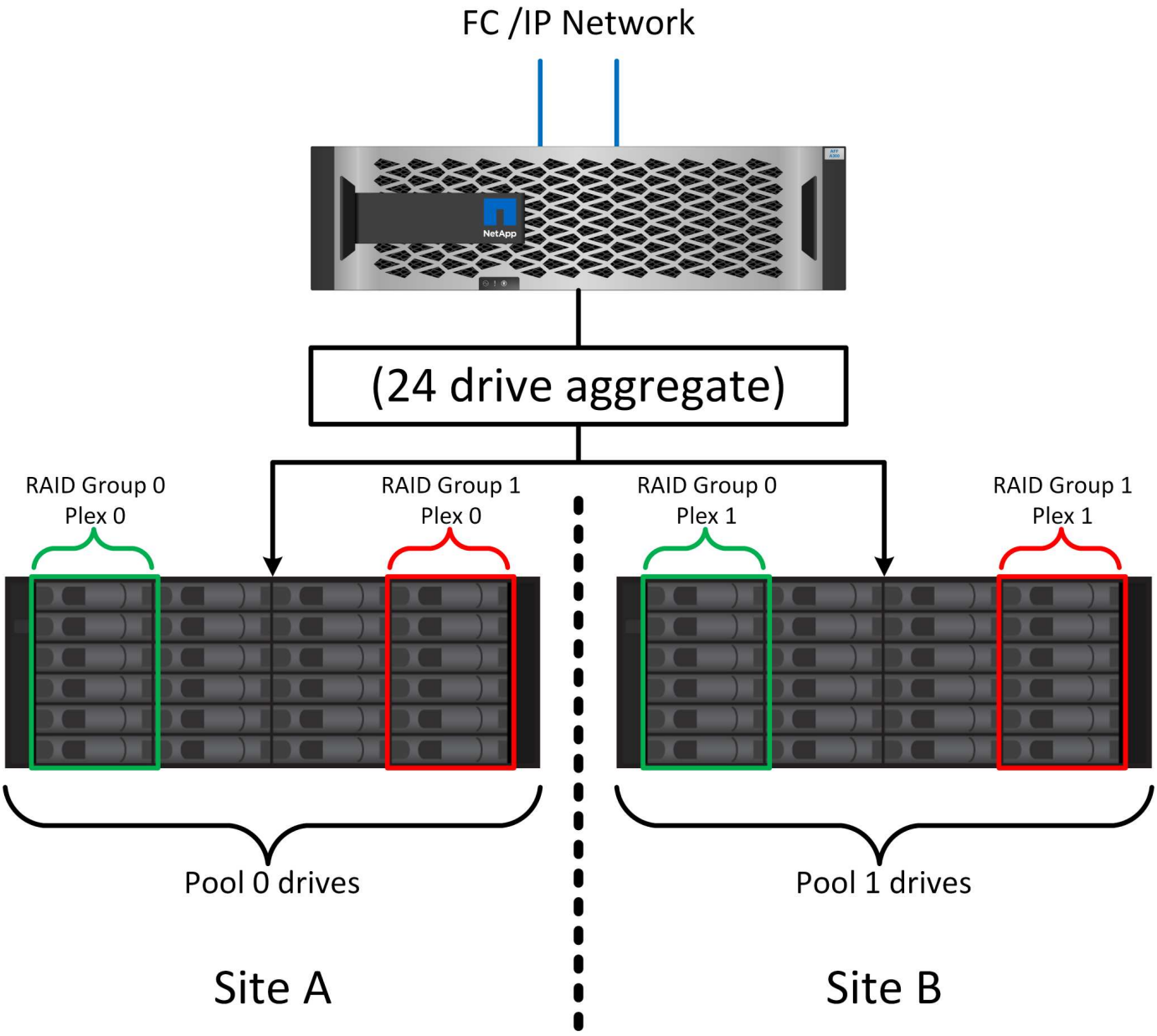
次の手順では、管理者がホストを完全にシャットダウンしてから、LUNとボリュームを手動で再度オンラインに戻します。これらの手順にはいくつかの作業が含まれる可能性がありますが、このアプローチはデータの整合性を確保するための最も安全な方法です。すべてのデータがこの保護を必要とするわけではありません。そのため、NVFAILの動作はボリューム単位で設定できます。

サイトおよびシェルフ障害からの保護：**SyncMirror**とプレックス

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。これにより、サイトが失われても継続的なデータ保護が実現します。



上の図は、SyncMirror構成の例を示しています。24ドライブのアグリゲートをコントローラに作成しました。このアグリゲートは、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本のドライブで構成されています。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、MetroClusterシステムに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、全体的には板金をほとんど使用していませんが、場合によっては追加の保護が保証されることがあります。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。

チエツクサム

チェックサムのトピックは、Oracle RMANのストリーミングバックアップをSnapshotベースのバックアップに移行することに慣れているDBAにとって特に関心があります。RMANの機能の1つは、バックアップ処理中に整合性チェックを実行することです。この機能には何らかの価値がありますが、その最大のメリットは、データベースが最新のストレージアレイで使用されていないことです。Oracleデータベースに物理ドライブが使用されている場合、ドライブの使用年数が経つと最終的にはほぼ確実に破損します。この問題は、真のストレージアレイではアレイベースのチェックサムによって解決されます。

実際のストレージアレイでは、複数のレベルでチェックサムを使用してデータの整合性が保護されます。IPベースのネットワークでデータが破損した場合、Transmission Control Protocol（TCP）レイヤはパケットデータを拒否し、再送信を要求します。FCプロトコルには、カプセル化されたSCSIデータと同様にチェックサムが含まれます。アレイに配置されたONTAPは、RAIDとチェックサムによる保護を備えています。破損は発生する可能性がありますが、ほとんどのエンタープライズアレイと同様に検出されて修正されます。通常、ドライブ全体に障害が発生してRAIDのリビルドが要求され、データベースの整合性は影響を受けません。ドライブ上の個 々 のバイトが宇宙線やフラッシュセルの故障によって損傷を受ける可能性があります。この場合、パリティチェックが失敗し、ドライブが故障し、RAIDのリビルドが開始されます。繰り返しになりますが、データの整合性には影響はありません。最後の防御線はチェックサムの使用です。たとえば、ドライブの致命的なファームウェアエラーが原因でRAIDパリティチェックで検出されなかった方法でデータが破損した場合、チェックサムは一致しません。ONTAPは破損したブロックをOracleデータベースが受信する前に転送を阻止します。

OracleのデータファイルとRedoログのアーキテクチャも、極度の状況下でも可能な限り最高レベルのデータ整合性を提供するように設計されています。最も基本的なレベルでは、Oracleのブロックにはチェックサムが含まれており、ほぼすべてのI/Oについて基本的な論理チェックが実行されます。Oracleがクラッシュしたり表領域がオフラインになったりしていない場合、データはそのまま維持されます。データ整合性チェックの程度は調整可能で、書き込みを確認するようにOracleを設定することもできます。その結果、クラッシュや障害のほぼすべてのシナリオをリカバリでき、非常にまれにリカバリ不能な状況が発生した場合は、破損がすぐに検出されます。

Oracleデータベースを使用しているNetAppのお客様のほとんどは'スナップショット・ベースのバックアップに移行すると'RMANなどのバックアップ製品の使用を中止しますRMANを使用してSnapCenterでブロックレベルのリカバリを実行できるオプションはまだあります。ただし、日常的には、RMAN、NetBackup、およびその他の製品は、月次または四半期ごとのアーカイブコピーの作成にのみ使用されます。

お客様の中には、 `dbv` 既存のデータベースの整合性チェックを定期的に実行します。NetAppでは、不必要なI/O負荷が発生するため、この方法は推奨されません。前述したように、データベースに以前に問題が発生していなかった場合、 `dbv` 問題の検出はほぼゼロです。このユーティリティは、ネットワークおよびストレージシステムに非常に高いシーケンシャルI/O負荷を生成します。Oracleの既知のバグにさらされるなど、破損が存在すると信じる理由がないかぎり、 `dbv`。

バックアップとリカバリの基本

Snapshotベースのバックアップ

ONTAPでのOracleデータベースのデータ保護の基盤となるのが、NetAppのSnapshotテクノロジです。

主な値は次のとおりです。

- *簡易性。*スナップショットは、特定の時点におけるデータのコンテナの内容の読み取り専用コピーです。
- 効率性。 Snapshotは作成時にスペースを必要としません。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。*スナップショットをベースにしたバックアップ戦略は、ストレージOSに標準で組み込まれているため、構成と管理が容易です。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セットで保護できます。
- ボリュームに1024個のSnapshotが含まれているかどうかに関係なく、パフォーマンスに影響はありません。

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内のSnapshotテクノロジは他に類を見ないものであり、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。

- 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。これにより、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、管理が簡易化されます。
- Snapshotコピーはパフォーマンスには影響しません。ただし、Snapshotに大量のデータが格納され、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が整合グループのバックアップを構成します。

また、ONTAPスナップショットは、競合するテクノロジよりも拡張性に優れています。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の最大数は1024です。Snapshotの保持期間を延長する必要がある場合は、Snapshot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でホストされているデータセットの保護はシンプルで拡張性に優れています。バックアップはデータの移動を必要としないため、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バックアップ戦略を調整できます。

Snapshotはバックアップですか？

データ保護戦略としてSnapshotを使用する場合によく寄せられる質問の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。

これは有効な問題です。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が唯一のバックアップ戦略であるべきではありません。そのため、NetAppは、SnapMirrorやSnapVaultレプリケーションなどのテクノロジを提供し、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します。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。

Snapshotベースのバックアップ

ONTAP Snapshotコピーを使用してデータを保護する方法は多数ありますが、Snapshotは、レプリケーション、ディザスタリカバリ、クローニングなど、ONTAPの他の多くの機能の基盤となります。Snapshotテクノロジの完全な概要については本ドキュメントでは説明しませんが、ここでは概要について説明します。

データセットのスナップショットを作成するには、主に次の2つの方法があります。

- crash-consistentバックアップ
- アプリケーションと整合性のあるバックアップ

データセットのcrash-consistentバックアップとは、ある時点におけるデータセット構造全体のキャプチャです。データセットが単一のボリュームに格納されている場合は、Snapshotはいつでも作成できるため、このプロセスは簡単です。データセット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（CG）Snapshotを作成する必要があります。CG Snapshotを作成するには、NetApp SnapCenterソフトウェア、ONTAPのネイティブ整合グループ機能、ユーザが管理するスクリプトなど、いくつかのオプションがあります。

crash-consistentバックアップは、主にpoint-of-the-backupリカバリで十分な場合に使用します。よりきめ細かなリカバリが必要な場合は、通常、アプリケーションと整合性のあるバックアップが必要です。

「application-consistent」の「consistent」という言葉は、しばしば誤った名義である。たとえば、Oracleデータベースをバックアップモードにすることをアプリケーション整合性バックアップと呼びますが、データの整合性が確保されたり休止されたりすることはありません。バックアップ中もデータは変化し続けます。一方、ほとんどのMySQLおよびMicrosoft SQL Serverのバックアップでは、バックアップを実行する前にデータが休止されます。VMwareは、特定のファイルの整合性を確保する場合としない場合があります。

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かった場合は、リストアを実行できません。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、整合性は常に基本的な機能であるため、ONTAPについて説明する際に「整合グループ」という用語はあまり使用されません。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。その後、データ保護を目的とした「整合グループ」として設定することもできますが、これは追加の設定手順です。

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、通常、正式には整合グループと呼ばれませんが、それが整合グループです。このボリュームのSnapshotは整合グループのイメージであり、そのSnapshotのリストアは整合グループのリストアです。SnapMirrorと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。

整合性グループの**Snapshot**

整合グループSnapshot（cg-snapshots）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-snapshots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おり、書き込み順序を維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-snapshots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によっては、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しなければならない場合があります。ワークフローの自動化や、cg-snapshot APIの呼び出しにカスタムスクリプトを使用することもできます。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。

一連のcg-snapshotsの作成は、次の2つの手順で行います。

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順番に確立されます。つまり、フェンシングプロセスが複数のボリュームにまたがって設定されている間は、最初のボリュームで書き込みI/Oがフリーズされ、以降に表示されるボリュームにコミットされ続けます。これは、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、データベースでは大量の非同期データファイル更新が問題され、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場合があります。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、フェンシングされたボリュームにあるファイルへの参照を削除するI/Oを実行すると、ファイルシステムのアクティビティが一時停止します。これにより、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フリーズされるため、Snapshotを正確に同時に作成する必要はありません。cg-snapshotを作成するアプリケーションの欠陥を防ぐために、初期の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。

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なく、実際の書き込み順序はオペレーティングシステムや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

ステムの更新も、書き込み順序に依存した書き込みの例です。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。

戦略

Snapshotベースのバックアップには、主に次の2つの方法があります。

- crash-consistentバックアップ
- Snapshotで保護されたホットバックアップ

データベースのcrash-consistentバックアップとは、データファイル、REDOログ、制御ファイルなど、データベース構造全体をある時点でキャプチャすることです。データベースが単一のボリュームに格納されている場合は、Snapshotはいつでも作成できるため、このプロセスは簡単です。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、整合性グループ（CG）Snapshotを作成する必要があります。CG Snapshotを作成するには、NetApp SnapCenterソフトウェア、ONTAPのネイティブ整合グループ機能、ユーザが管理するスクリプトなど、いくつかのオプションがあります。

crash-consistent Snapshotバックアップは、主にポイントオブザバックアップリカバリで十分な場合に使用されます。状況によってはアーカイブログを適用できますが、よりきめ細かなポイントインタイムリカバリが必要な場合は、オンラインバックアップを推奨します。

Snapshotベースのオンラインバックアップの基本的な手順は次のとおりです。

1. データベースを `backup` モード（Mode）：
2. データファイルをホストしているすべてのボリュームのSnapshotを作成します。
3. 終了します `backup` モード（Mode）：
4. コマンドを実行します `alter system archive log current` ログのアーカイブを強制的に実行します。
5. アーカイブログをホストするすべてのボリュームのSnapshotを作成します。

この手順により、バックアップモードのデータファイルと、バックアップモード中に生成された重要なアーカイブログを含む一連のSnapshotが作成されます。データベースのリカバリには、次の2つの要件があります。制御ファイルなどのファイルも便宜上保護する必要がありますが、絶対に必要なのはデータファイルとアーカイブログの保護だけです。

戦略はお客様によって大きく異なる可能性がありますが、これらの戦略のほとんどは、最終的には以下に概説されているのと同じ原則に基づいています。

Snapshotベースのリカバリ

Oracleデータベースのボリュームレイアウトを設計する際には、ボリュームベースNetApp SnapRestore（VBSR）テクノロジを使用するかどうかを最初に決定します。

ボリュームベースのSnapRestoreを使用すると、ボリュームをある時点の状態にほぼ瞬時にリバートできます。VBSRはボリューム上のすべてのデータがリバートされるため、すべてのユースケースに適しているとは限りません。たとえば、データファイル、Redoログ、アーカイブログを含むデータベース全体が1つのボリュームに格納されている場合、このボリュームをVBSRでリストアすると、新しいアーカイブログとRedoデータが破棄されるためデータが失われます。

リストアにVBSRは必要ありません。データベースの多くは、ファイルベースのSingle-File SnapRestore（SFSR）を使用するか、Snapshotからアクティブファイルシステムにファイルをコピーして戻すだけでリス

トアできます。

VBSRは、データベースが非常に大規模な場合やできるだけ迅速にリカバリする必要がある場合に推奨されます。また、VBSRを使用するにはデータファイルを分離する必要があります。NFS環境では、特定のデータベースのデータファイルを、他の種類のファイルの影響を受けない専用ボリュームに格納する必要があります。SAN環境では、データファイルを専用ボリュームの専用LUNに格納する必要があります。ボリュームマネージャを使用する場合は（Oracle Automatic Storage Management[ASM]を含む）、ディスクグループもデータファイル専用にする必要があります。

この方法でデータファイルを分離すると、他のファイルシステムに影響を与えることなく、データファイルを以前の状態にリバートできます。

Snapshot リザーブ

SAN環境内のOracleデータを含むボリュームごとに、 `percent-snapshot-space` LUN環境でSnapshot用にスペースをリザーブしても役に立たないため、ゼロに設定する必要があります。フラクショナルリザーブを100に設定すると、LUNを含むボリュームのSnapshotでは、すべてのデータの書き替えを100%吸収するために、Snapshotリザーブを除くボリューム内に十分な空きスペースが必要になります。フラクショナルリザーブの値を小さい値に設定すると、それに応じて必要な空きスペースは少なくなりますが、Snapshotリザーブは常に除外されます。これは、LUN環境のスナップショット予約スペースが無駄になることを意味します。

NFS環境には2つのオプションがあります。

- を設定します `percent-snapshot-space` 予想されるSnapshotスペース消費量に基づきます。
- を設定します `percent-snapshot-space` アクティブなスペース使用量とSnapshotスペース使用量をまとめてゼロにして管理できます。

最初のオプションでは、 `percent-snapshot-space` は、ゼロ以外の値（通常は約20%）に設定されます。このスペースはユーザーには表示されません。ただし、この値によって利用率が制限されるわけではありません。リザーブが20%のデータベースで30%の入れ替えが発生した場合、スナップショット領域は20%リザーブの範囲を超えて拡張され、リザーブされていないスペースを占有する可能性があります。

リザーブを20%などの値に設定する主な利点は、一部のスペースが常にスナップショットに使用可能であることを確認することです。たとえば、1TBのボリュームに20%のリザーブが設定されている場合、データベース管理者（DBA）が格納できるのは800GBのデータのみです。この構成では、Snapshot用に少なくとも200GBのスペースが保証されます。

いつ `percent-snapshot-space` がゼロに設定されている場合、ボリューム内のすべてのスペースをエンドユーザが使用できるため、可視性が向上します。データベース管理者は、Snapshotを利用する1TBのボリュームが表示された場合、この1TBのスペースはアクティブデータとSnapshotの書き替えの間で共有されることを理解しておく必要があります。

エンドユーザ間では、オプション1とオプション2の間に明確な優先順位はありません。

ONTAPとサードパーティのスナップショット

Oracle Doc ID 604683.1には、サードパーティ製スナップショットのサポート要件と、バックアップおよびリストア処理に使用できる複数のオプションが説明されています。

サードパーティベンダーは、会社のスナップショットが次の要件に準拠していることを保証する必要があります。

- スナップショットは、Oracleが推奨するリストアおよびリカバリ処理と統合する必要があります。

- スナップショットは、スナップショットの時点でデータベースクラッシュ整合性がある必要があります。
- スナップショット内のファイルごとに書き込み順序が保持されます。

ONTAPおよびNetAppのOracle管理製品は、これらの要件に準拠しています。

SnapRestore

NetApp SnapRestoreテクノロジは、SnapshotからのONTAPでのデータの高速リストアを実現します。

重要なデータセットが使用できないと、重要なビジネス処理が停止します。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する場合でも、ネットワーク上での転送に時間がかかることがあります。SnapRestoreでは、データセット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には、ファイル/ LUNベースとボリュームベースの2つの形式があります。

- 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。
- ファイルやLUNのコンテナは、10GBでも100TBのデータでも、数秒でリストアできます。

「ファイルまたはLUNのコンテナ」とは、通常はFlexVolボリュームを指します。たとえば、1つのボリューム内に1つのLVMディスクグループを構成する10個のLUNを配置したり、1つのボリュームに1、000ユーザのNFSホームディレクトリを格納したりできます。個 々 のファイルまたはLUNに対してリストア処理を実行する代わりに、ボリューム全体を単一の処理としてリストアできます。このプロセスは、FlexGroupやONTAP整合グループなど、複数のボリュームを含むスケールアウトコンテナとも連携します。

SnapRestoreがこれほど迅速かつ効率的に機能するのは、Snapshotの性質によるものです。Snapshotは本質的には、特定の時点におけるボリュームの内容を読み取り専用で並行して表示する機能です。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は、Snapshot作成時のファイルおよび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。スナップショットはデータの読み取り/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**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、API処理やCLI処理の処理には数秒かかることがありますが、リストアプロセスは基本的に瞬時に完了します。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、多くの企業のお客様が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、これには以降に作成されたスナップショットが含まれることはあまり明白ではありません。これは必ずしも望ましいとは限りません。

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SnapRestoreを使用して5日前に作成されたSnapshotにデータセットをリストアすると、過去5日間に作成されたSnapshotがすべ

て破棄され、SLAに違反します。

この制限に対処するために、いくつかのオプションが用意されています。

1. ボリューム全体のSnapRestoreを実行するのではなく、以前のSnapshotからデータをコピーできます。この方法は、データセットが小さい場合に最も適しています。
2. Snapshotはリストアではなくクローニングできます。このアプローチの制限事項は、ソーススナップショットがクローンの依存関係であることです。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の使用。

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。スナップショットを削除する必要はありません。また、この操作によって以前のスナップショットへ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リストア中に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、この処理が完了するまでSnapshotの作成はブロックされます。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。

オンラインハツクアツフ

バックアップモードでOracleデータベースを保護およびリカバリするには、2セットのデータが必要です。これはOracleの唯一のバックアップ・オプションではなく’最も一般的なバックアップ・オプションであることに注意してください

- バックアップモードでのデータファイルのSnapshot
- データファイルが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ログのセット

オンライン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。これは、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、すべてのリストア手順が熟練したデータベース管理者によって処理される大規模な本番環境では特に顕著です。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。

一部の大規模なお客様では、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ます。たとえば、次のコマンドをスケジュールします。 `alter database begin backup` 23時58分、`alter database end backup` 00:02に実行し、午前0時にストレージシステム上でSnapshotの直接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では、専用ボリューム内でのデータファイルの分離についても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配置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。オンライン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が `spfile` および `passwd` データファイルをホストしているディスクグループにファイルがありません。これにより、データファイルのみを選択してリストアすることができなくなります。

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必要があります。リストアされていない場合は、RMAN / sqlplusをsnapshotディレクトリ内のデータに転送できます。

また、小規模なデータベースの場合は、エンドユーザがデータファイルを `.snapshot` 自動化ツールやストレージ管理者の支援がないディレクトリで、 `snaprestore` コマンドを実行します

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. 目的のリストアポイントの直前に、データファイルディスクグループをSnapshotにリストアします。
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。この例では、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、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。

ストレージ**Snapshot**で最適化されたバックアップ

Oracle 12cがリリースされた時点では、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はさらにシンプルになりました。そのため、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しても、完全なリカバリやポイントインタイムリカバリを引き続き実行できます。

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可能でした。Oracle 10gおよび11gでは、データベースの整合性を維持するために、リカバリ時に手動で追加の手順を実行する必要がありました。Oracle 12cでは、 `sqlplus` および `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でアーカイブログを再生するための追加ロジックが含まれています。

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。

ストレージSnapshotを最適化したリカバリでは、同じ結果を達成するために必要なデータセットがわずかに異なります。

- データファイルのSnapshot、およびSnapshot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Snapshot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するためです。

SANでは、専用ボリューム内でのデータファイルの分離についても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配置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] ASM spfileファイルとpasswd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ないことを確認します。これにより、データファイルのみを選択してリストアすることができなくなります。

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必要があります。または、 `rman` または `sqlplus` のデータに転送できます。 `.snapshot` ディレクトリ。

また、小規模なデータベースの場合は、エンドユーザがデータファイルを `.snapshot` SnapRestoreコマンドを実行するための自動化ツールやストレージ管理者の支援がないディレクトリ。

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. 目的のリストアポイントの直前に、データファイルディスクグループをSnapshotにリストアします。
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。この例では、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@host1 ~]$ 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1つのコマンドで実行できます。 `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。

```
Cluster01::> snapshot show -vserver vserver1 -volume NTAP_oradata -fields
create-time
vserver   volume        snapshot   create-time
--------  ------------  ---------  ------------------------
vserver1  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@host1 ~]$ 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>
```

Snapshotを使用してデータベースを完全にリカバリするには、 `recover automatic` コマンドには特定のライセンスは不要ですが、を使用してポイントインタイムリカバリを実行できます。 `snapshot time` Oracle Advanced Compressionのライセンスが必要です。

データベース管理と自動化のためのツール

Oracle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。

これらのコア機能をONTAPに直接簡単に設定して要件を満たす場合もありますが、より複雑なニーズにはオーケストレーションレイヤが必要です。

SnapCenter

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製品に似ていますが、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構築されています。

SnapCenterには、Snapshotベースのバックアップとリストア、SnapMirrorとSnapVaultのレプリケーションな

ど、大企業の大規模な運用に必要な基本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上のSnapCenterプラグインの無停止での一元管理、クラウド規模環境向けに設計されたユーザインターフェイスなどがあります。

REST

ONTAPには、豊富なRESTful APIセットも含まれています。これにより、サードパーティベンダーは、ONTAPとの緊密な統合により、データ保護やその他の管理アプリケーションを作成できます。さらに、独自の自動化ワークフローやユーティリティを作成したいお客様も、RESTful APIを簡単に利用できます。

Oracleのディザスタリカバリ

概要

ディザスタリカバリとは、火災によってストレージシステムやサイト全体が破壊されるなど、重大な災害が発生した場合にデータサービスをリストアすることです。

このドキュメントは、以前に公開されたテクニカルレポート_TR-4591：『Oracle Data Protection_and_TR-4592：Oracle on MetroCluster』を差し替えます。_

ディザスタリカバリは、もちろんSnapMirrorを使用してデータを単純にレプリケーションすることで実現できます。多くのお客様は、ミラーされたレプリカを1時間に何度も更新します。

ほとんどのお客様にとって、DRに必要なのはデータのリモートコピーだけではなく、そのデータを迅速に利用できることです。NetAppは、このニーズに対応する2つのテクノロジを提供します。MetroClusterとSnapMirrorのアクティブ同期です。

MetroClusterとは、低レベルの同期ミラーリングストレージと多数の追加機能を含むハードウェア構成のONTAPのことです。MetroClusterなどの統合ソリューションは、今日の複雑なスケールアウトデータベース、アプリケーション、仮想化インフラストラクチャを簡素化します。複数の外部データ保護製品や戦略を、1つのシンプルな中央集中型ストレージアレイに置き換えます。また、単一のクラスタストレージシステム内に、バックアップ、リカバリ、ディザスタリカバリ、高可用性（HA）が統合されています。

SnapMirrorアクティブ同期（SM-AS）はSnapMirror同期に基づいています。MetroClusterでは、各ONTAPコントローラがドライブデータをリモートサイトにレプリケートします。SnapMirrorアクティブ同期を使用すると、基本的には2つの異なるONTAPシステムでLUNデータの独立したコピーを維持しながら、このLUNの単一インスタンスを提供できます。ホストの観点からは、単一のLUNエンティティです。

SM-ASとMCCの比較

SM-ASとMetroClusterは全体的な機能が似ていますが、RPO=0レプリケーションの実装方法と管理方法には重要な違いがあります。SnapMirrorの非同期および同期はDR計画の一部としても使用できますが、HAレプリケーションテクノロジとしては設計されていません。

- MetroCluster構成は、複数のサイトにノードが分散された統合クラスタのようなものです。SM-ASは、同期的にレプリケートされるRPO=0のLUNにサービスを提供する独立した2つのクラスタのように動作します。
- MetroCluster構成のデータには、常に1つの特定のサイトからしかアクセスできません。データの2つ目のコピーは反対側のサイトに存在しますが、データはパッシブです。ストレージシステムのフェイルオーバーがないとアクセスできません。

- MetroClusterとSM-ASによるミラーリングは、さまざまなレベルで実行されます。MetroClusterミラーリングはRAIDレイヤで実行されます。下位レベルのデータは、SyncMirrorを使用してミラーリングされた形式で格納されます。ミラーリングは、LUN、ボリューム、プロトコルの各レイヤでは実質的に使用されません。
- 一方、SM-ASミラーリングはプロトコルレイヤで行われます。2つのクラスタは、全体的に独立したクラスタです。データの2つのコピーが同期されると、2つのクラスタは書き込みをミラーリングするだけで済みます。一方のクラスタで書き込みが発生すると、もう一方のクラスタにレプリケートされます。書き込みの確認応答がホストに送信されるのは、両方のサイトで書き込みが完了した場合だけです。このプロトコルスプリット動作以外では、2つのクラスタは通常のONTAPクラスタです。
- MetroClusterの主な役割は大規模なレプリケーションです。RPO=0でRTOがほぼゼロのアレイ全体をレプリケートできます。フェイルオーバーが1つしかなく、容量とIOPSの点で非常に適切に拡張できるため、フェイルオーバープロセスが簡易化されます。
- SM-ASの主なユースケースの1つに、きめ細かなレプリケーションがあります。すべてのデータを1つのユニットとしてレプリケートしたくない場合や、特定のワークロードを選択的にフェイルオーバーできる必要がある場合があります。
- SM-ASのもう1つの主なユースケースは、アクティブ/アクティブ処理です。アクティブ/アクティブ処理では、データの完全に使用可能なコピーを、同じパフォーマンス特性を持つ2つの異なるクラスタに配置し、必要に応じてSANをサイト間で拡張する必要がありません。アプリケーションを両方のサイトで実行しておくことで、フェイルオーバー処理中の全体的なRTOを短縮できます。

MetroCluster

MetroClusterによるディザスタリカバリ

MetroClusterは、サイト間のRPO=0の同期ミラーリングでOracleデータベースを保護するONTAPの機能です。また、単一のMetroClusterシステムで数百のデータベースをサポートするまでスケールアップできます。

使い方も簡単です。MetroClusterを使用しても、エンタープライズアプリケーションやデータベースの運用に最適な条件が追加されたり変更されたりするとは限りません。

通常のベストプラクティスも引き続き適用され、必要なデータ保護がRPO=0の場合はMetroClusterで対応します。しかし、ほとんどのお客様は、RPO=0のデータ保護だけでなく、災害時のRTOを向上させ、サイトメンテナンス作業の一環として透過的なフェイルオーバーを実現するためにMetroClusterを使用しています。

物理アーキテクチャ

MetroCluster環境でのOracleデータベースの動作を理解するには、MetroClusterシステムの物理設計についてある程度の説明が必要です。

このドキュメントは、以前に公開されていたテクニカルレポート（TR-4592：『Oracle on MetroCluster』）に代わるものです。_

MetroClusterは**3**種類の構成で使用できます。

- IPセツソクノHAヘア
- FCセツソクノHAヘア
- シングルコントローラ、FC接続

「接続」という用語は、サイト間レプリケーションに使用されるクラスタ接続を指します。ホストプロトコルを指しているわけではありません。MetroCluster構成では、クラスタ間通信に使用される接続の種類に関係なく、すべてのホスト側プロトコルが通常どおりサポートされます。

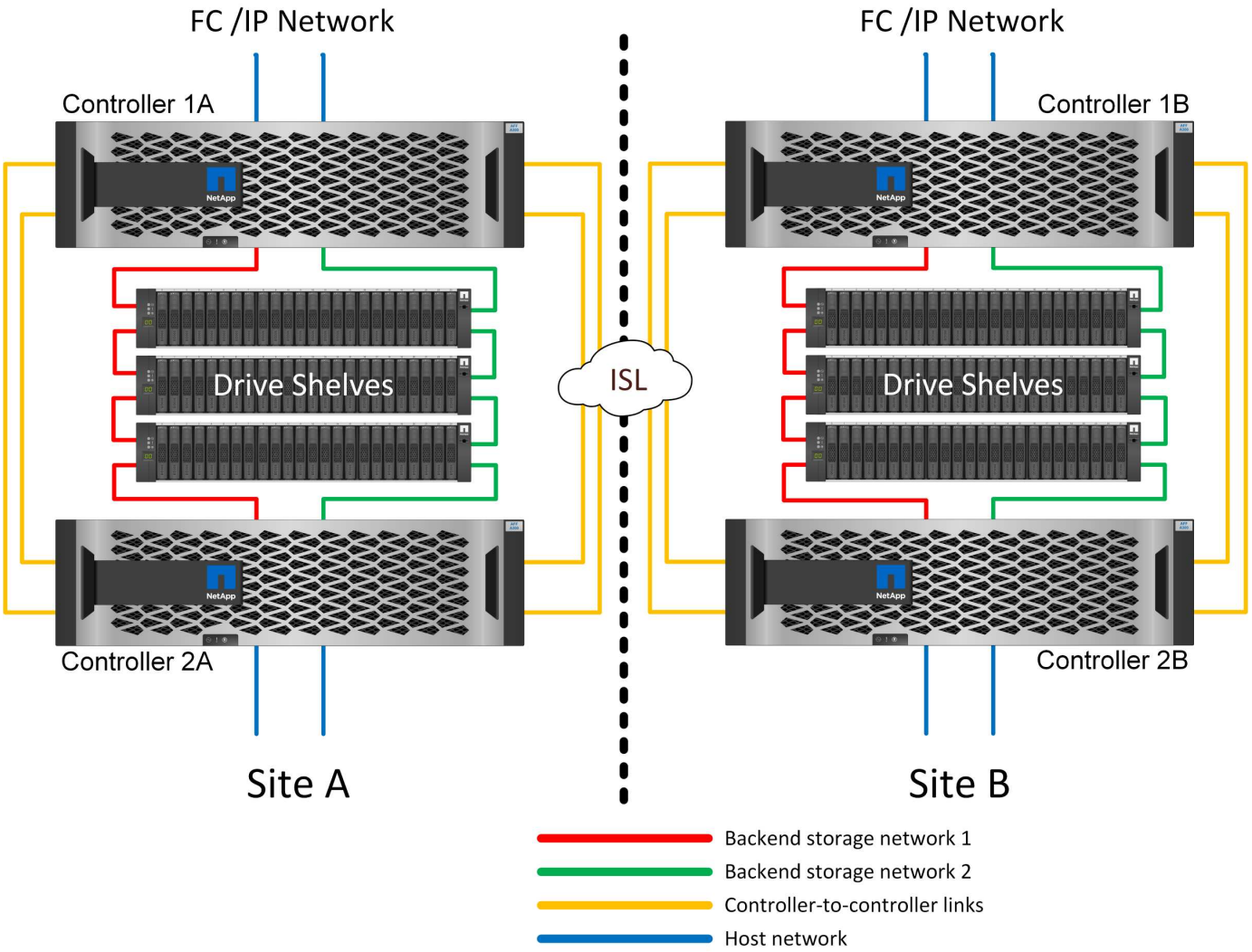
MetroCluster IP の略

HAペアMetroCluster IP構成では、サイトごとに2ノードまたは4ノードを使用します。この設定オプションを使用すると、2ノードオプションに比べて複雑さとコストが増加しますが、サイト内の冗長性という重要なメリットがあります。単純なコントローラ障害では、WAN経由のデータアクセスは必要ありません。データアクセスは、代替ローカルコントローラを介してローカルのままです。

ほとんどのお客様は、インフラストラクチャの要件がシンプルであるため、IP接続を選択しています。これまでは、ダークファイバやFCスイッチを使用した場合、サイト間での高速接続のプロビジョニングは一般的に容易でしたが、今日では、高速で低レイテンシのIP回線がより容易に利用可能になっています。

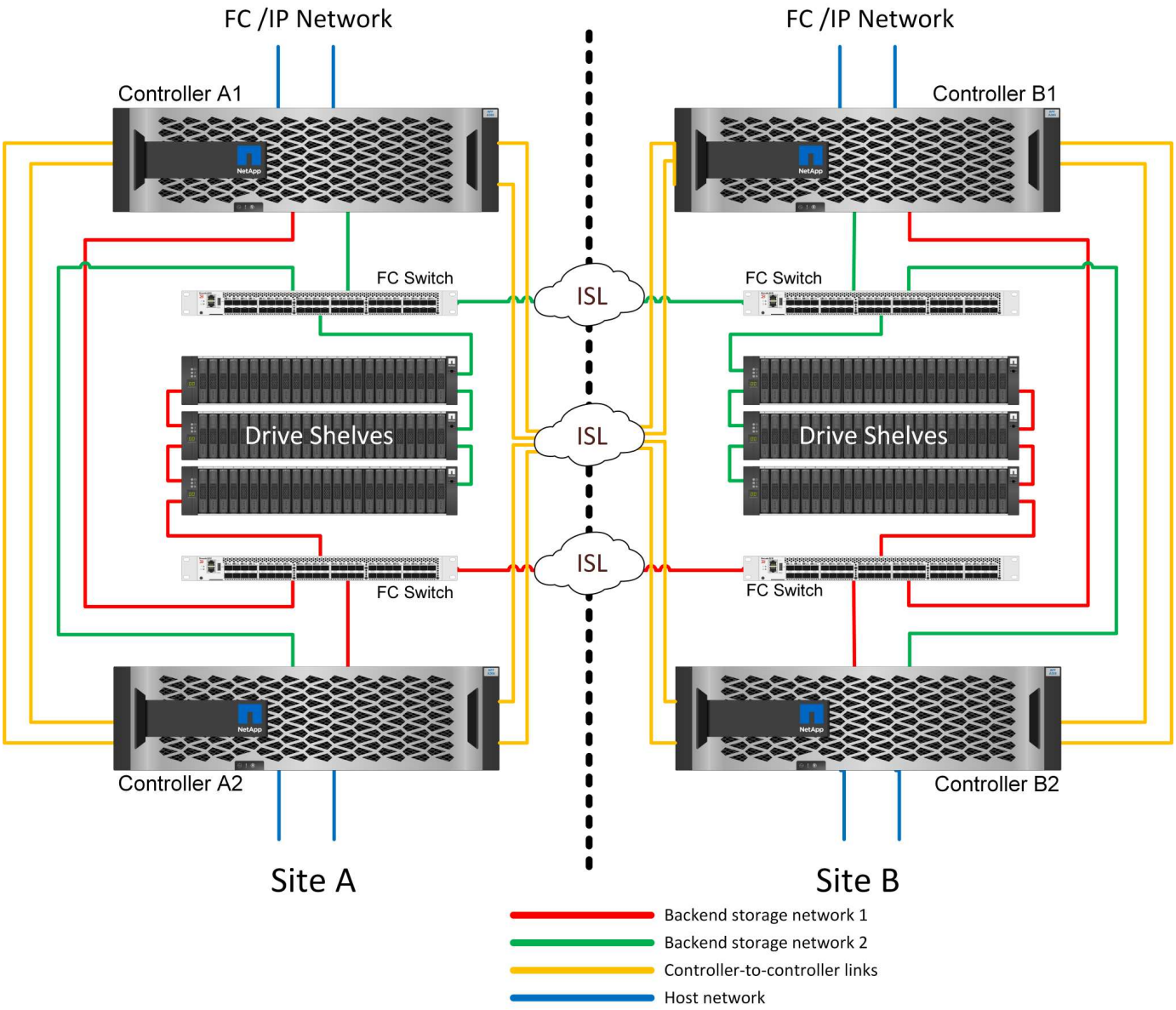
サイト間接続はコントローラのみであるため、アーキテクチャもシンプルです。FC SAN接続MetroClusterでは、コントローラが反対側サイトのドライブに直接書き込むため、追加のSAN接続、スイッチ、およびブリッジが必要になります。一方、IP構成のコントローラは、コントローラを介して反対側のドライブに書き込みます。

追加情報については、ONTAPの公式ドキュメントを参照してください。 "MetroCluster IP 解決策のアーキテクチャと設計"。



HAペアFC SAN接続MetroCluster

HAペアMetroCluster FC構成では、サイトごとに2ノードまたは4ノードを使用します。この設定オプションを使用すると、2ノードオプションに比べて複雑さとコストが増加しますが、サイト内の冗長性という重要なメリットがあります。単純なコントローラ障害では、WAN経由のデータアクセスは必要ありません。データアクセスは、代替ローカルコントローラを介してローカルのままです。

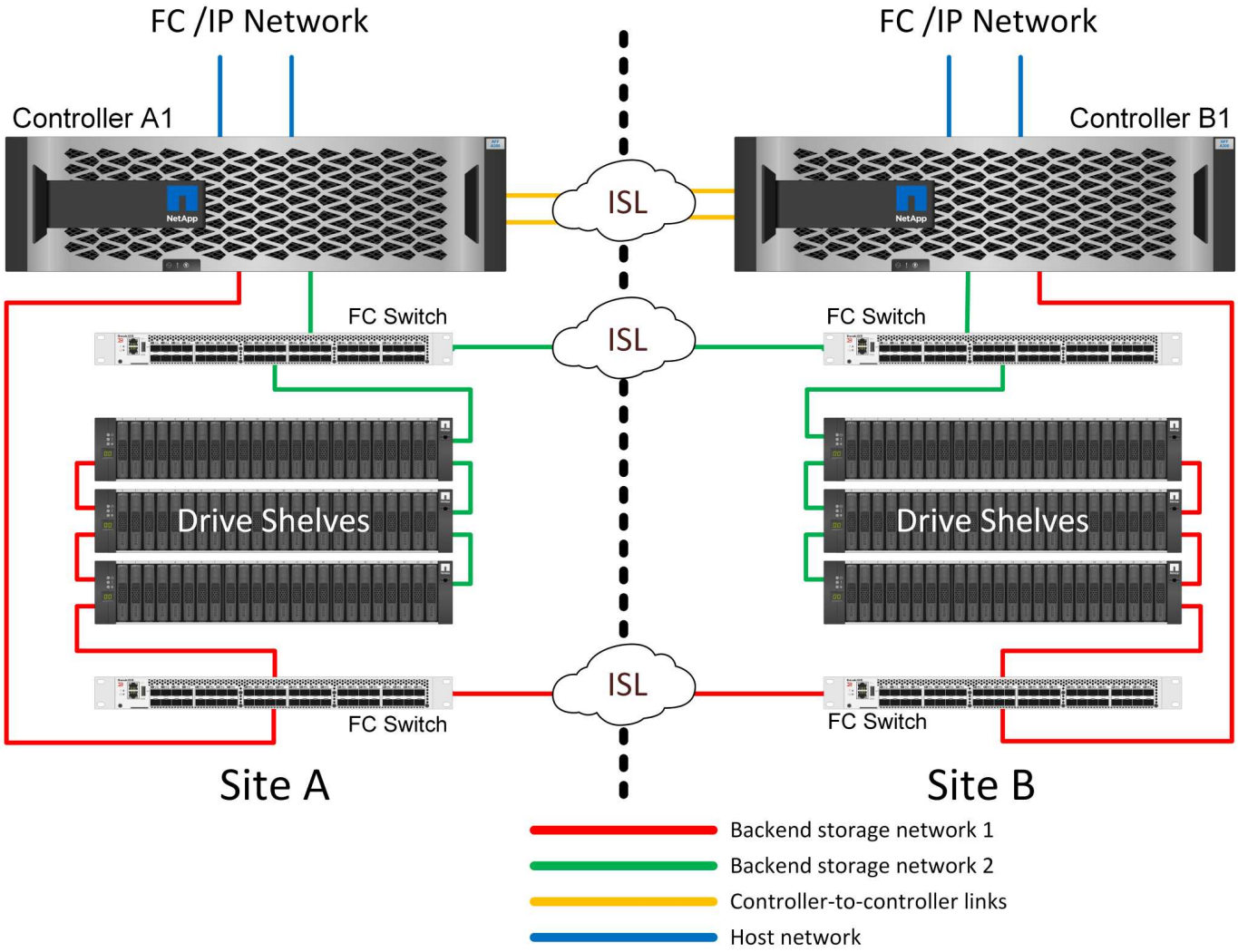


一部のマルチサイトインフラは、アクティブ/アクティブ運用向けに設計されたものではなく、プライマリサイトやディザスタリカバリサイトとして使用されます。この場合、一般にHAペアMetroClusterオプションが推奨される理由は次のとおりです。

- 2ノードMetroClusterクラスタはHAシステムですが、コントローラに予期しない障害が発生した場合や計画的メンテナンスを行う場合は、反対側のサイトでデータサービスをオンラインにする必要があります。サイト間のネットワーク接続が必要な帯域幅をサポートできない場合は、パフォーマンスが低下します。唯一の選択肢は、さまざまなホストOSと関連サービスを代替サイトにフェイルオーバーすることです。HAペアMetroClusterクラスタでは、コントローラが停止すると同じサイト内で単純なフェイルオーバーが発生するため、この問題は解消されます。
- 一部のネットワークトポロジは、サイト間アクセス用に設計されていませんが、異なるサブネットまたは分離されたFC SANを使用します。この場合、代替コントローラが反対側のサイトのサーバにデータを提供できないため、2ノードMetroClusterクラスタはHAシステムとして機能しなくなります。完全な冗長性を実現するには、HAペアMetroClusterオプションが必要です。
- 2サイトインフラを単一の高可用性インフラとみなす場合は、2ノードMetroCluster構成が適しています。ただし、サイト障害後もシステムが長時間機能しなければならない場合は、HAペアが推奨されます。HAペアは、単一サイト内でHAを提供し続けるためです。

2ノードFC SAN接続MetroCluster

2ノードMetroCluster構成では、サイトごとに1つのノードのみが使用されます。設定とメンテナンスが必要なコンポーネントが少ないため、HAペアオプションよりもシンプルな設計になっています。また、ケーブル配線やFCスイッチの点でインフラストラクチャの必要性も軽減されています。最後に、コストを削減します。



この設計の明らかな影響は、1つのサイトでコントローラに障害が発生した場合、反対側のサイトからデータを利用できることです。この制限は必ずしも問題ではありません。多くの企業は、本質的に単一のインフラとして機能する、拡張された高速で低レイテンシのネットワークを使用したマルチサイトデータセンター運用を行っています。このような場合は、2ノードバージョンのMetroClusterが推奨されます。2ノードシステムは現在、複数のサービスプロバイダでペタバイト規模で使用されています。

MetroClusterの耐障害性機能

MetroCluster 解決策 には単一点障害（Single Point of Failure）はありません。

- 各コントローラに、ローカルサイトのドライブシェルフへの独立したパスが2つあります。
- 各コントローラに、リモートサイトのドライブシェルフへの独立したパスが2つあります。
- 各コントローラには、反対側のサイトのコントローラへの独立したパスが2つあります。
- HAペア構成では、各コントローラからローカルパートナーへのパスが2つあります。

つまり、構成内のコンポーネントを1つでも削除しても、MetroClusterのデータ提供機能を損なうことはありません。2つのオプションの耐障害性の違いは、サイト障害後もHAペアバージョンが全体的なHAストレージシステムになる点だけです。

論理アーキテクチャ

MetroCluster環境でOracleデータベースがどのように動作するかを理解するAlsopでは、MetroClusterシステムの論理機能について説明する必要があります。

サイト障害からの保護：**NVRAM**と**MetroCluster**

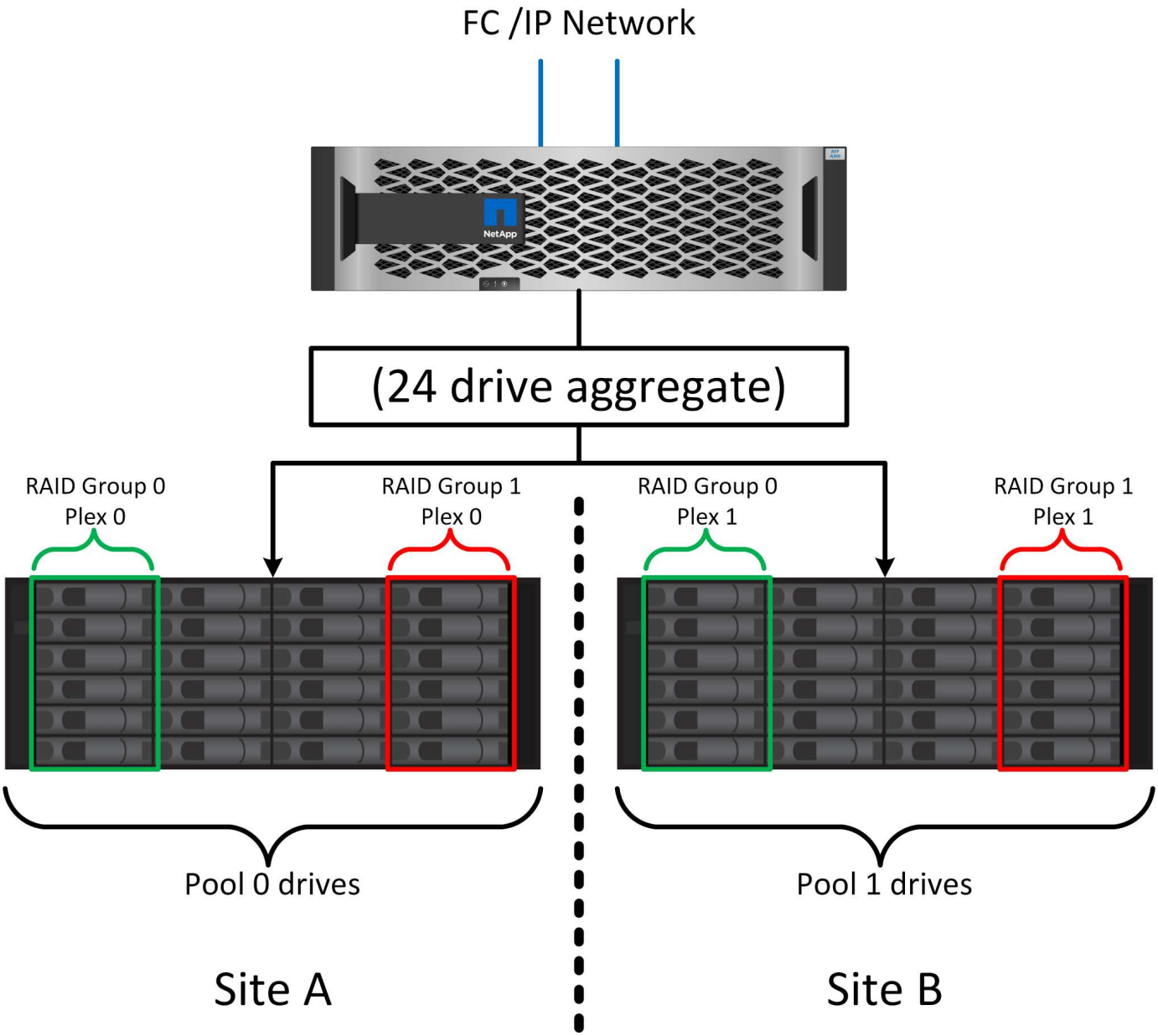
MetroClusterは、次の方法でNVRAMデータ保護を拡張します。

- 2ノード構成では、NVRAMデータがスイッチ間リンク（ISL）を使用してリモートパートナーにレプリケートされます。
- HAペア構成では、NVRAMデータがローカルパートナーとリモートパートナーの両方にレプリケートされます。
- 書き込みは、すべてのパートナーにレプリケートされるまで確認応答されません。このアーキテクチャは、NVRAMデータをリモートパートナーにレプリケートすることで、転送中のI/Oをサイト障害から保護します。このプロセスは、ドライブレベルのデータレプリケーションには関係ありません。アグリゲートを所有するコントローラは、アグリゲート内の両方のプレックスに書き込むことでデータレプリケーションを実行しますが、サイトが失われた場合でも転送中のI/Oの損失からデータを保護する必要があります。レプリケートされたNVRAMデータは、障害が発生したコントローラをパートナーコントローラがテイクオーバーする必要がある場合にのみ使用されます。

サイトおよびシェルフ障害からの保護：**SyncMirror**とプレックス

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。

1. 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。
2. 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。
3. サイト上の各ドライブセットは、ミラーリングを使用せずに、完全に冗長化された1つ以上のRAID DPグループまたはRAID-TECグループとして自動的に構成されます。ミラーリングの下でRAIDを使用することで、サイトが失われた場合でもデータを保護できます。



上の図は、SyncMirror構成の例を示しています。24ドライブのアグリゲートをコントローラに作成しました。このアグリゲートは、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本のドライブで構成されています。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ドライブのプレックスが含まれており、サイトBの6ドライブの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ドライブのプレックスが含まれており、サイトBの6ドライブのプレックスにミラーリングされています。

SyncMirrorは通常、MetroClusterシステムに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、全体的には板金をほとんど使用していませんが、場合によっては追加の保護が保証されることがあります。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、独立した電源供給と独立したUPSシステムを備えた2つの物理ラックに分かれていました。

前述したように、書き込みの確認応答は、少なくとも1台の他のコントローラでローカルのNVRAMとNVRAMに記録されるまで返されません。このアプローチにより、ハードウェア障害や停電が発生しても、転送中のI/Oが失われることはありません。ローカルのNVRAMに障害が発生したり、他のノードへの接続に障害が発生したりすると、データはミラーリングされなくなります。

ローカルNVRAMからエラーが報告されると、ノードはシャットダウンします。このシャットダウンにより、HAペアが使用されている場合はパートナーコントローラにフェイルオーバーされます。MetroClusterでは、動作は選択した全体的な設定によって異なりますが、リモートノートに自動的にフェイルオーバーされる場合があります。いずれの場合も、障害が発生したコントローラが書き込み処理を認識していないため、データは失われません。

リモートノードへのNVRAMレプリケーションがブロックされるサイト間接続障害は、より複雑な状況です。書き込みがリモートノードにレプリケートされなくなるため、コントローラで重大なエラーが発生した場合にデータが失われる可能性があります。さらに重要なことは、このような状況で別のノードにフェイルオーバーしようとするとデータが失われることです。

制御要素は、NVRAMが同期されているかどうかです。NVRAMが同期されていれば、ノード間のフェイルオーバーを安全に実行でき、データ損失のリスクはありません。MetroCluster構成では、NVRAMと基盤となるアグリゲートのプレックスが同期されていれば、データ損失のリスクなしにスイッチオーバーを実行できます。

データが同期されていない場合、ONTAPは、フェイルオーバーまたはスイッチオーバーを強制的に実行しないかぎり、フェイルオーバーまたはスイッチオーバーを許可しません。この方法で条件を変更すると、元のコントローラにデータが残っている可能性があり、データ損失が許容されることが確認されます。

データベースやその他のアプリケーションは、ディスク上のデータのより大きな内部キャッシュを保持するため、フェイルオーバーやスイッチオーバーを強制的に実行した場合に特に破損の影響を受けやすくなります。強制的なフェイルオーバーまたはスイッチオーバーが発生した場合、以前に確認済みの変更は事実上破棄されます。ストレージアレイの内容は実質的に時間を逆方向にジャンプし、キャッシュの状態はディスク上のデータの状態を反映しなくなります。

この状況を回避するために、ONTAPでは、NVRAMの障害に対する特別な保護をボリュームに設定できます。この保護メカニズムがトリガーされると、ボリュームがNVFAILという状態になります。この状態になると、原因アプリケーションがクラッシュするI/Oエラーが発生します。このクラッシュにより、古いデータを使用しないようにアプリケーションがシャットダウンされます。コミットされたトランザクションデータがログに含まれている必要があるため、データが失われないようにしてください。次の手順では、管理者がホストを完全にシャットダウンしてから、LUNとボリュームを手動で再度オンラインに戻します。これらの手順にはいくつかの作業が含まれる可能性がありますが、このアプローチはデータの整合性を確保するための最も安全な方法です。すべてのデータがこの保護を必要とするわけではありません。そのため、NVFAILの動作はボリューム単位で設定できます。

HAペアと**MetroCluster**

MetroClusterには、2ノードとHAペアの2つの構成があります。2ノード構成の動作は、NVRAMに関してはHAペアと同じです。突然の障害が発生した場合、パートナーノードはNVRAMデータを再生してドライブの整合性を確保し、確認済みの書き込みが失われていないことを確認できます。

HAペア構成では、ローカルパートナーノードにもNVRAMがレプリケートされます。MetroClusterを使用しないスタンドアロンHAペアの場合と同様に、単純なコントローラ障害ではパートナーノードでNVRAMが再生されます。サイト全体が突然失われた場合、リモートサイトには、ドライブの整合性を確保してデータの提供を開始するために必要なNVRAMも用意されています。

MetroClusterの重要な側面の1つは、通常の運用状態ではリモートノードがパートナーデータにアクセスできないことです。各サイトは本質的に、反対のサイトのパーソナリティを想定できる独立したシステムとして機能します。このプロセスはスイッチオーバーと呼ばれ、計画的スイッチオーバーでは、サイトの処理が無停止で反対側のサイトに移行されます。また、サイトが失われ、ディザスタリカバリの一環として手動または自動のスイッチオーバーが必要になる計画外の状況も含まれます。

スイッチオーバーとスイッチバック

スイッチオーバーとスイッチバックという用語は、MetroCluster構成のリモートコントローラ間でボリュームを移行するプロセスを指します。このプロセスでは、リモートノードのみが環境されます。4ボリューム構成でMetroClusterを使用する場合のローカルノードのフェイルオーバーは、前述したテイクオーバーとギブバックのプロセスと同じです。

計画的スイッチオーバーとスイッチバック

計画的スイッチオーバーまたはスイッチバックは、ノード間のテイクオーバーやギブバックと似ています。このプロセスには複数の手順があり、数分かかるように見える場合もありますが、実際には、ストレージリソースとネットワークリソースを複数のフェーズで正常に移行します。完全なコマンドの実行に必要な時間よりもはるかに短時間で制御転送が行われる瞬間。

テイクオーバー/ギブバックとスイッチオーバー/スイッチバックの主な違いは、FC SAN接続への影響です。ローカルのテイクオーバー/ギブバックでは、ローカルノードへのFCパスがすべて失われ、ホストのネイティブMPIOを使用して使用可能な代替パスに切り替えます。ポートは再配置されません。スイッチオーバーとスイッチバックでは、コントローラの仮想FCターゲットポートがもう一方のサイトに移行します。一時的にSAN上に存在しなくなり、代わりのコントローラに再表示されます。

SyncMirrorタイムアウト

SyncMirrorは、シェルフ障害から保護するONTAPのミラーリングテクノロジです。シェルフが離れた場所に配置されている場合は、リモートデータ保護が実現します。

SyncMirrorは汎用同期ミラーリングを提供しません。その結果、可用性が向上します。一部のストレージシステムでは、一定のオールオアナッシングミラーリング（Dominoモードと呼ばれることもあります）を使用します。リモートサイトへの接続が失われるとすべての書き込みアクティビティが停止する必要があるため、この形式のミラーリングはアプリケーションで制限されます。そうしないと、書き込みは一方のサイトに存在し、もう一方のサイトには存在しません。通常、このような環境では、サイト間の接続が短時間（30秒など）以上切断された場合にLUNがオフラインになるように構成されます。

この動作は、一部の環境に適しています。ただし、ほとんどのアプリケーションには、通常の動作条件下で保証された同期レプリケーションを提供しながら、レプリケーションを一時停止できる解決策が必要です。サイト間の接続が完全に失われると、多くの場合、災害が近い状況とみなされます。通常、このような環境は、接続が修復されるか、データを保護するために環境をシャットダウンする正式な決定が下されるまで、オンラインのままでデータを提供します。リモートレプリケーションの障害のみが原因でアプリケーションを自動的にシャットダウンする必要があるのは珍しいことです。

SyncMirrorは、タイムアウトの柔軟性を備えた同期ミラーリングの要件に対応しています。リモートコントローラやプレックスへの接続が失われると、30秒のタイマーがカウントダウンを開始します。カウンタが0に達すると、ローカルデータを使用して書き込みI/O処理が再開されます。データのリモートコピーは使用可能ですが、接続が回復するまで時間内に凍結されます。再同期では、アグリゲートレベルのSnapshotを使用してシステムをできるだけ迅速に同期モードに戻します。

特に、多くの場合、この種の汎用的なオールオアナッシングDominoモードレプリケーションは、アプリケーションレイヤでより適切に実装されています。たとえば、Oracle DataGuardには最大保護モードが用意され

ており、どのような状況でも長時間のインスタンスレプリケーションが保証されます。設定可能なタイムアウトを超えてレプリケーションリンクに障害が発生すると、データベースはシャットダウンします。

ファブリック接続**MetroCluster**による自動無人スイッチオーバー

Automatic Unattended Switchover（AUSO；自動無人スイッチオーバー）は、クロスサイトHAの形式を提供するファブリック接続MetroClusterの機能です。前述したように、MetroClusterには2つのタイプ（各サイトに1台のコントローラを配置する場合と、各サイトに1台のHAペアを配置する場合）があります。HAオプションの主な利点は、コントローラの計画的シャットダウンと計画外シャットダウンのどちらでもすべてのI/Oをローカルで処理できることです。シングルノードオプションのメリットは、コスト、複雑さ、インフラの削減です。

AUSOの主な価値は、ファブリック接続MetroClusterシステムのHA機能を向上させることです。各サイトが反対側のサイトの健常性を監視し、データを提供するノードがなくなると、AUSOによって迅速なスイッチオーバーが実行されます。このアプローチは、可用性の点でHAペアに近い構成になるため、サイトごとにノードが1つだけのMetroCluster構成で特に役立ちます。

AUSOでは、HAペアレベルで包括的な監視を行うことはできません。HAペアには、ノード間の直接通信用の2本の冗長な物理ケーブルが含まれているため、きわめて高い可用性を実現できます。さらに、HAペアの両方のノードが冗長ループ上の同じディスクセットにアクセスできるため、1つのノードが別のノードの健常性を監視するための別のルートが提供されます。

MetroClusterクラスタは複数のサイトにまたがって存在し、ノード間の通信とディスクアクセスの両方がサイト間ネットワーク接続に依存します。クラスタの残りの部分のハートビートを監視する機能には制限があります。AUSOは、ネットワークの問題が原因で、もう一方のサイトが使用できない状況ではなく、実際にダウンしている状況を区別する必要があります。

その結果、HAペアのコントローラで、システムパニックなどの特定の理由で発生したコントローラ障害が検出された場合、テイクオーバーが要求されることがあります。また、接続が完全に失われた場合（ハートビートの損失とも呼ばれます）、テイクオーバーを促すこともあります。

MetroClusterシステムで自動スイッチオーバーを安全に実行できるのは、元のサイトで特定の障害が検出された場合のみです。また、ストレージシステムの所有権を取得するコントローラは、ディスクとNVRAMのデータが同期されていることを保証できる必要があります。コントローラは、ソースサイトとの通信が失われて稼働している可能性があるため、スイッチオーバーの安全性を保証できません。スイッチオーバーを自動化するためのその他のオプションについては、次のセクションのMetroCluster Tiebreaker（MCTB）解決策に関する情報を参照してください。

ファブリック接続**MetroCluster**を使用した**MetroCluster Tiebreaker**

この"NetApp MetroCluster Tiebreaker"ソフトウェアを第3のサイトで実行すると、MetroCluster環境の健全性を監視し、通知を送信できます。また、災害時にオプションでスイッチオーバーを強制的に実行することもできます。Tiebreakerの詳細についてはを参照して"NetApp Support Site"ください。MetroCluster Tiebreakerの主な目的はサイトの損失を検出することです。また、サイトの損失と接続の損失を区別する必要があります。たとえば、Tiebreakerがプライマリサイトに到達できなかったためにスイッチオーバーが発生しないようにします。そのため、Tiebreakerはリモートサイトがプライマリサイトに接続する能力も監視します。

AUSOによる自動スイッチオーバーもMCTBと互換性があります。AUSOは、特定の障害イベントを検出し、NVRAMとSyncMirrorのプレックスが同期されている場合にのみスイッチオーバーを実行するように設計されているため、非常に迅速に対応します。

一方、Tiebreakerはリモートに配置されているため、サイトの停止を宣言する前にタイマーが経過するのを待つ必要があります。Tiebreakerは最終的にAUSOの対象となるコントローラ障害を検出しますが、一般的にはAUSOがスイッチオーバーを開始しており、Tiebreakerが機能する前にスイッチオーバーを完了している可

能性があります。Tiebreakerから送信される2つ目のswitchoverコマンドは拒否されます。

> MCTBソフトウェアは、強制的なスイッチオーバー時に、NVRAM WASまたはプレックス（あるいはその両方）が同期されていることを検証しません。メンテナンス作業中に自動スイッチオーバーが設定されている場合は無効にして、NVRAMまたはSyncMirrorプレックスの同期が失われるようにしてください。

また、MCTBは、次の一連のイベントにつながるローリングディザスタに対応できない場合があります。

1. サイト間の接続が30秒以上中断されます。
2. SyncMirrorレプリケーションがタイムアウトし、プライマリサイトで処理が続行されるため、リモートレプリカは古くなります。
3. プライマリサイトが失われます。その結果、プライマリサイトにレプリケートされていない変更が存在します。その場合、次のようないくつかの理由でスイッチオーバーが望ましくない可能性があります。
   - 重要なデータはプライマリサイトに存在し、最終的にリカバリ可能になる可能性があります。スイッチオーバーによってアプリケーションの動作が継続されると、重要なデータは実質的に破棄されます。
   - サバイバーサイトのアプリケーションで、サイト障害時にプライマリサイトのストレージリソースを使用していた場合、データがキャッシュされている可能性があります。スイッチオーバーでは、キャッシュと一致しない古いバージョンのデータが生成されます。
   - サバイバーサイトのオペレーティングシステムで、サイト障害時にプライマリサイトのストレージリソースを使用していた場合、キャッシュデータがある可能性があります。スイッチオーバーでは、キャッシュと一致しない古いバージョンのデータが生成されます。最も安全な方法は、Tiebreakerがサイト障害を検出した場合にアラートを送信するように設定し、スイッチオーバーを強制的に実行するかどうかを決定することです。キャッシュされたデータを消去するには、アプリケーションやオペレーティングシステムのシャットダウンが必要になる場合があります。さらに、NVFAIL設定を使用して保護を強化し、フェイルオーバープロセスを合理化することもできます。

MetroCluster IPを使用した**ONTAP**メディエーター

ONTAPメディエーターは、MetroCluster IPおよびその他の特定のONTAPソリューションで使用されます。これは、前述のMetroCluster Tiebreakerソフトウェアと同様に従来のTiebreakerサービスとして機能しますが、重要な機能を実行する自動無人スイッチオーバーも含まれています。

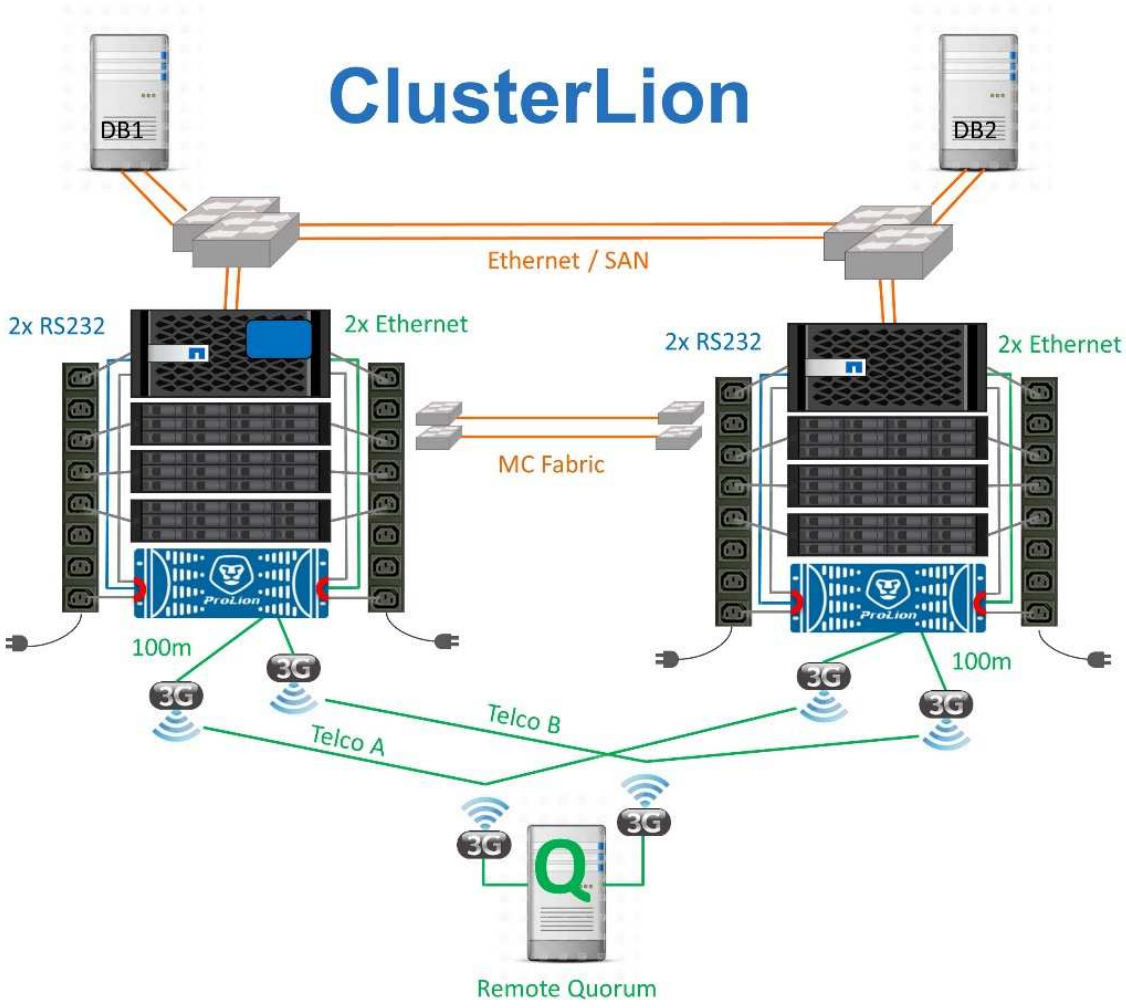
ファブリック接続MetroClusterは、反対側のサイトのストレージデバイスに直接アクセスできます。これにより、一方のMetroClusterコントローラがドライブからハートビートデータを読み取ることで、他のコントローラの健常性を監視できます。これにより、一方のコントローラがもう一方のコントローラの障害を認識し、スイッチオーバーを実行できるようになります。

一方、MetroCluster IPアーキテクチャでは、すべてのI/Oがコントローラとコントローラの接続を介して排他的にルーティングされるため、リモートサイトのストレージデバイスに直接アクセスすることはありません。これにより、コントローラで障害を検出してスイッチオーバーを実行する機能が制限されます。そのため、サイトの損失を検出して自動的にスイッチオーバーを実行するためには、ONTAPメディエーターがTiebreakerデバイスとして必要になります。

ClusterLionを使用した**3**番目の仮想サイト

ClusterLionは、仮想の第3サイトとして機能する高度なMetroCluster監視アプライアンスです。このアプローチにより、完全に自動化されたスイッチオーバー機能により、MetroClusterを2サイト構成で安全に導入できます。さらに、ClusterLionでは、追加のネットワークレベル監視を実行し、スイッチオーバー後の処理を実行

できます。完全なドキュメントはProLionから入手できます。



- ClusterLionアプライアンスは、直接接続されたイーサネットケーブルとシリアルケーブルでコントローラの健常性を監視します。
- 2つのアプライアンスは、冗長3Gワイヤレス接続で相互に接続されています。
- ONTAPコントローラへの電源は、内部リレーを介して配線されます。サイト障害が発生すると、内部UPSシステムを搭載したClusterLionによって電源接続が切断されてからスイッチオーバーが実行されます。このプロセスにより、スプリットブレイン状態が発生しないようにします。
- ClusterLionは、30秒のSyncMirrorタイムアウト内にスイッチオーバーを実行するか、まったく実行しません。
- ClusterLionでは、NVRAMプレックスとSyncMirrorプレックスの状態が同期されていないかぎり、スイッチオーバーは実行されません。
- ClusterLionでは、MetroClusterが完全に同期されている場合にのみスイッチオーバーが実行されるため、NVFAILは必要ありません。この構成では、計画外スイッチオーバーが発生しても、拡張Oracle RACなどのサイトスパニング環境をオンラインのまま維持できます。
- ファブリック接続MetroClusterとMetroCluster IPの両方をサポート

SyncMirror

MetroClusterシステムを使用したOracleデータ保護の基盤となるのは、最大パフォーマンスのスケールアウト同期ミラーリングテクノロジであるSyncMirrorです。

SyncMirrorによるデータ保護

最も簡単な意味では、同期レプリケーションとは、変更がミラーされたストレージの両側に対して確認応答される前に行われなければならないことを意味します。たとえば、データベースがログを書き込んでいる場合やVMwareゲストにパッチを適用している場合は、書き込みが失われることはありません。プロトコルレベルでは、両方のサイトの不揮発性メディアにコミットされるまで、ストレージシステムは書き込みを確認応答しないでください。その場合にのみ、データ損失のリスクなしに作業を安全に進めることができます。

同期レプリケーションテクノロジの使用は、同期レプリケーション解決策を設計および管理するための最初のステップです。最も重要な考慮事項は、計画的および計画外のさまざまな障害シナリオで何が発生するかを理解することです。すべての同期レプリケーションソリューションが同じ機能を提供するわけではありません。Recovery Point Objective（RPO；目標復旧時点）がゼロ（つまりデータ損失ゼロ）の解決策が必要な場合は、すべての障害シナリオを考慮する必要があります。特に、サイト間の接続が失われてレプリケーションが不可能になった場合、どのような結果が予想されますか。

SyncMirrorデータの可用性

MetroClusterレプリケーションは、同期モードに効率的に切り替えられるように設計されたNetApp SyncMirrorテクノロジに基づいています。この機能は、同期レプリケーションを必要とする一方で、データサービスに高可用性も必要とするお客様の要件を満たします。たとえば、リモートサイトへの接続が切断されている場合は、通常、ストレージシステムをレプリケートされていない状態で運用し続けることを推奨します。

多くの同期レプリケーションソリューションは、同期モードでしか動作できません。このタイプのall-or-nothingレプリケーションは、Dominoモードと呼ばれることがあります。このようなストレージシステムでは、データのローカルコピーとリモートコピーが非同期になるのではなく、データの提供が停止します。レプリケーションが強制的に解除された場合、再同期には非常に時間がかかり、ミラーリングの再確立中にデータが完全に失われる可能性があります。

リモートサイトに到達できない場合にSyncMirrorを同期モードからシームレスに切り替えることができるだけでなく、接続がリストアされたときにRPO=0状態に迅速に再同期することもできます。再同期中にリモートサイトにある古いデータコピーを使用可能な状態で保持することもできるため、データのローカルコピーとリモートコピーを常に維持できます。

Dominoモードが必要な場合、NetAppはSnapMirror Synchronous（SM-S）を提供します。Oracle DataGuardやSQL Server Always On可用性グループなど、アプリケーションレベルのオプションも用意されています。オプションとして、OSレベルのディスクミラーリングを使用できます。追加情報とオプションについては、担当のNetAppまたはパートナーアカウントチームにお問い合わせください。

MetroClusterとNVFAIL

NVFailはONTAPの一般的なデータ整合性機能で、データベースを使用してデータ整合性を最大限に保護するように設計されています。

このセクションでは、基本的なONTAP NVFAILについて説明し、MetroCluster固有のトピックを扱います。

MetroClusterでは、少なくとも1台の他のコントローラのローカルNVRAMとNVRAMに書き込みが記録される

まで、書き込み確認は行われません。このアプローチにより、ハードウェア障害や停電が発生しても、転送中のI/Oが失われることはありません。ローカルのNVRAMに障害が発生したり、他のノードへの接続に障害が発生したりすると、データはミラーリングされなくなります。

ローカルNVRAMからエラーが報告されると、ノードはシャットダウンします。このシャットダウンにより、HAペアが使用されている場合はパートナーコントローラにフェイルオーバーされます。MetroClusterでは、動作は選択した全体的な設定によって異なりますが、リモートノートに自動的にフェイルオーバーされる場合があります。いずれの場合も、障害が発生したコントローラが書き込み処理を認識していないため、データは失われません。

リモートノードへのNVRAMレプリケーションがブロックされるサイト間接続障害は、より複雑な状況です。書き込みがリモートノードにレプリケートされなくなるため、コントローラで重大なエラーが発生した場合にデータが失われる可能性があります。さらに重要なことは、このような状況で別のノードにフェイルオーバーしようとするとデータが失われることです。

制御要素は、NVRAMが同期されているかどうかです。NVRAMが同期されていれば、ノード間のフェイルオーバーを安全に実行でき、データ損失のリスクはありません。MetroCluster構成では、NVRAMと基盤となるアグリゲートのプレックスが同期されていれば、データ損失のリスクなしにスイッチオーバーを安全に実行できます。

データが同期されていない場合、ONTAPは、フェイルオーバーまたはスイッチオーバーを強制的に実行しないかぎり、フェイルオーバーまたはスイッチオーバーを許可しません。この方法で条件を変更すると、元のコントローラにデータが残っている可能性があり、データ損失が許容されることが確認されます。

データベースは、ディスク上のデータのより大きな内部キャッシュを保持するため、フェイルオーバーやスイッチオーバーを強制的に実行した場合、データベースが破損する可能性が特に高くなります。強制的なフェイルオーバーまたはスイッチオーバーが発生した場合、以前に確認済みの変更は事実上破棄されます。ストレージアレイの内容は実質的に時間を逆方向に移動し、データベースキャッシュの状態はディスク上のデータの状態を反映しなくなります。

この状況からアプリケーションを保護するために、ONTAPでは、NVRAMの障害に対する特別な保護をボリュームに設定できます。この保護メカニズムがトリガーされると、ボリュームがNVFAILという状態になります。この状態になると、古いデータを使用しないように原因アプリケーションをシャットダウンするI/Oエラーが発生します。確認済みの書き込みはストレージシステムに残っているため、データが失われることはありません。データベースの場合は、コミットされたトランザクションデータがログに含まれている必要があります。

次の手順では、管理者がホストを完全にシャットダウンしてから、LUNとボリュームを手動で再度オンラインに戻します。これらの手順にはいくつかの作業が含まれる可能性がありますが、このアプローチはデータの整合性を確保するための最も安全な方法です。すべてのデータがこの保護を必要とするわけではありません。そのため、NVFAILの動作はボリューム単位で設定できます。

手動強制**NVFAIL**

サイト間に分散されているアプリケーションクラスタ（VMware、Oracle RACなど）でスイッチオーバーを強制的に実行する最も安全なオプションは、 `-force-nvfail-all` コマンドラインで。このオプションは、キャッシュされたすべてのデータが確実にフラッシュされるようにするための緊急措置として使用できます。災害が発生したサイトにもともと配置されていたストレージリソースをホストが使用している場合、I/Oエラーまたは古いファイルハンドルのいずれかを受信します。 (`ESTALE`）エラー。Oracleデータベースがクラッシュし、ファイルシステムが完全にオフラインになるか、読み取り専用モードに切り替わります。

スイッチオーバーの完了後、 `in-nvfailed-state` フラグをクリアし、LUNをオンラインにする必要があります。このアクティビティが完了したら、データベースを再起動できます。これらのタスクを自動化してRTOを短縮できます。

dr-force-nvfail

一般的な安全対策として、 `dr-force-nvfail` 通常運用時にリモートサイトからアクセスされる可能性があるすべてのボリューム（フェイルオーバー前に使用されるアクティビティ）にフラグを付けます。この設定により、選択したリモートボリュームが `in-nvfailed-state` スイッチオーバー中。スイッチオーバーの完了後、 `in-nvfailed-state` フラグをクリアし、LUNをオンラインにする必要があります。これらのアクティビティが完了したら、アプリケーションを再起動できます。これらのタスクを自動化してRTOを短縮できます。

結果は、 `-force-nvfail-all` 手動スイッチオーバーのフラグ。ただし、影響を受けるボリュームの数は、古いキャッシュを使用するアプリケーションまたはオペレーティングシステムから保護する必要があるボリュームだけに制限される場合があります。

を使用しない環境には、次の2つの重要な要件があります。 `dr-force-nvfail` アプリケーションボリューム：

- 強制スイッチオーバーは、プライマリサイトの障害から30秒以内に実行する必要があります。
- メンテナンスタスクの実行中や、SyncMirrorプレックスやNVRAMレプリケーションが同期されていないその他の状況では、スイッチオーバーを実行しないでください。最初の要件を満たすには、Tiebreakerソフトウェアを使用します。Tiebreakerソフトウェアは、サイト障害から30秒以内にスイッチオーバーを実行するように設定されています。これは、サイト障害が検出されてから30秒以内にスイッチオーバーを実行する必要があるという意味ではありません。これは、サイトが動作していることが確認されてから30秒が経過した場合に強制的にスイッチオーバーを実行しても安全ではないことを意味します。

2つ目の要件は、MetroCluster構成が同期されていないことが判明した場合に、自動スイッチオーバー機能をすべて無効にすることで部分的に満たすことができます。NVRAMレプリケーションとSyncMirrorプレックスの健常性を監視できるTiebreaker解決策を使用することを推奨します。クラスタが完全に同期されていない場合、Tiebreakerはスイッチオーバーをトリガーしません。

NetApp MCTBソフトウェアは同期ステータスを監視できないため、何らかの理由でMetroClusterが同期されていない場合は無効にする必要があります。ClusterLionにはNVRAM監視機能とプレックス監視機能が搭載されており、MetroClusterシステムが完全に同期されていることが確認されないかぎり、スイッチオーバーをトリガーしないように設定できます。

Oracleシングルインスタンス

前述したように、MetroClusterシステムが存在しても、データベースの運用に関するベストプラクティスが必ずしも追加されたり変更されたりするわけではありません。お客様のMetroClusterシステムで現在実行されているデータベースの大部分はシングルインスタンスであり、Oracle on ONTAPドキュメントに記載されている推奨事項に従っています。

事前設定された**OS**を使用したフェイルオーバー

SyncMirrorはディザスタリカバリサイトにデータの同期コピーを提供しますが、そのデータを利用できるようにするには、オペレーティングシステムと関連するアプリケーションが必要です。基本的な自動化により、環境全体のフェイルオーバー時間を大幅に短縮できます。Veritas Cluster Server（VCS）などのClusterware製品は、サイト間でクラスタを作成するためによく使用されます。多くの場合、フェイルオーバープロセスは単純なスクリプトで実行できます。

プライマリノードが失われた場合、代替サイトでデータベースをオンラインにするようにクラスタウェア（またはスクリプト）が設定されます。1つは、データベースを構成するNFSリソースまたはSANリソース用に事

前設定されたスタンバイサーバを作成する方法です。プライマリサイトに障害が発生すると、クラスタウェアまたはスクリプト化された代替サイトが次のような一連の処理を実行します。

1. MetroClusterスイッチオーバーの強制実行
2. FC LUNの検出の実行（SANのみ）
3. ファイルシステムのマウント、ASMディスクグループのマウント
4. データベースの起動

このアプローチの主な要件は、リモートサイトでOSを実行することです。Oracleバイナリを使用して事前に設定する必要があります。つまり、Oracleのパッチ適用などのタスクをプライマリサイトとスタンバイサイトで実行する必要があります。また、災害が発生した場合は、Oracleバイナリをリモートサイトにミラーリングしてマウントすることもできます。

実際のアクティベーション手順は簡単です。LUN検出などのコマンドでは、FCポートあたりのコマンド数が少なくて済みます。ファイル・システムのマウントは' `mount` コマンドを実行し、データベースとASMの両方をCLIで1つのコマンドで起動および停止できます。スイッチオーバーの前にディザスタリカバリサイトでボリュームとファイルシステムを使用していない場合は、 `dr-force- nvfail` ボリューム：

仮想**OS**によるフェイルオーバー

データベース環境のフェイルオーバーを拡張して、オペレーティングシステム自体を含めることができます。理論的には、このフェイルオーバーはブートLUNで実行できますが、ほとんどの場合、仮想OSで実行されます。手順の手順は次のようになります。

1. MetroClusterスイッチオーバーの強制実行
2. データベースサーバ仮想マシンをホストするデータストアのマウント
3. 仮想マシンの起動
4. データベースを手動で起動するか、データベースを自動的に起動するように仮想マシンを設定します。たとえば、ESXクラスタが複数のサイトにまたがっている場合があります。災害が発生した場合は、スイッチオーバー後にディザスタリカバリサイトで仮想マシンをオンラインにすることができます。災害発生時に仮想データベースサーバをホストするデータストアが使用されていないかぎり、 `dr-force- nvfail` 関連付けられているボリューム。

Oracle拡張**RAC**

多くのお客様が、Oracle RACクラスタを複数のサイトにまたがって構成し、完全なアクティブ/アクティブ構成を実現することで、RTOを最適化しています。Oracle RACのクォーラム管理を含める必要があるため、設計全体が複雑になります。また、データは両方のサイトからアクセスされるため、強制的スイッチオーバーによって古いデータコピーが使用される可能性があります。

データのコピーは両方のサイトに存在しますが、データを提供できるのはアグリゲートを現在所有しているコントローラだけです。そのため、拡張RACクラスタでは、リモートのノードがサイト間接続でI/Oを実行する必要があります。その結果、I/Oレイテンシが増加しますが、このレイテンシは一般的には問題になりません。RACインターコネクトネットワークは複数のサイトにまたがって拡張する必要があるため、とにかく高速で低レイテンシのネットワークが必要です。レイテンシが増加して原因に問題が発生した場合は、クラスタをアクティブ/パッシブで運用できます。I/O負荷の高い処理は、アグリゲートを所有するコントローラに対してローカルなRACノードに対して実行する必要があります。リモートノードは、より軽いI/O処理を実行するか、純粋にウォームスタンバイサーバとして使用されます。

アクティブ/アクティブ拡張RACが必要な場合は、MetroClusterの代わりにSnapMirrorアクティブ同期を検討する必要があります。SM-ASレプリケーションでは、データの特定のレプリカを優先的に使用できます。したがって、すべての読み取りがローカルに行われる拡張RACクラスタを構築できます。読み取りI/Oがサイトを経由することはないため、レイテンシは最小限に抑えられます。すべての書き込みアクティビティは引き続きサイト間接続を転送する必要がありますが、このようなトラフィックは同期ミラーリング解決策では回避できません。

仮想ブートディスクを含むブートLUNをOracle RACで使用する場合は、 `misscount`パラメータの変更が必要になることがあります。RACタイムアウトパラメータの詳細については、を参照してください"ONTAPを使用したOracle RAC"。

2サイト構成

2サイトの拡張RAC構成では、すべてではないが多くの災害シナリオに無停止で対応できるアクティブ/アクティブデータベースサービスを提供できます。

RAC投票ファイル

MetroClusterに拡張RACを導入する場合は、クォーラム管理を最初に検討する必要があります。Oracle RACには、クォーラムを管理するための2つのメカニズム（ディスクハートビートとネットワークハートビート）があります。ディスクハートビートは、投票ファイルを使用してストレージアクセスを監視します。単一サイトのRAC構成では、基盤となるストレージシステムがHA機能を提供していれば、単一の投票リソースで十分です。

以前のバージョンのOracleでは、投票ファイルは物理ストレージデバイスに配置されていましたが、現在のバージョンのOracleでは、投票ファイルはASMディスクグループに格納されていました。

Oracle RACはNFSでサポートされています。グリッドのインストールプロセスでは、一連のASMプロセスが作成され、グリッドファイルに使用されるNFSの場所がASMディスクグループとして提供されます。このプロセスはエンドユーザに対してほぼ透過的であり、インストール完了後にASMを継続的に管理する必要はありません。

2サイト構成の最初の要件は、無停止のディザスタリカバリプロセスを保証する方法で、各サイトが常に半数以上の投票ファイルにアクセスできるようにすることです。このタスクは、投票ファイルがASMディスクグループに格納される前は簡単でしたが、今日の管理者はASM冗長性の基本原則を理解する必要があります。

ASMディスクグループには3つの冗長性オプションがあります。 `external`、 `normal`および `high`。つまり、ミラーリングされていない、ミラーリングされている、3方向ミラーリングされているということです。という新しいオプションがあります。 `Flex` 利用可能ですが、めったに使用されません。冗長デバイスの冗長性レベルと配置によって、障害が発生した場合の動作が制御されます。例：

- 投票ファイルをに配置する `diskgroup` を使用 `external` 冗長性リソースを使用すると、サイト間接続が失われた場合に一方のサイトの削除が保証されます。
- 投票ファイルをに配置する `diskgroup` を使用 `normal` 各サイトにASMディスクが1つしかない冗長性を確保すると、どちらのサイトにもマジョリティクォーラムが存在しないためにサイト間接続が失われた場合に、両方のサイトでノードが削除されます。
- 投票ファイルをに配置する `diskgroup` を使用 `high` 一方のサイトに2本のディスクを配置し、もう一方のサイトに1本のディスクを配置する冗長性により、両方のサイトが動作していて相互にアクセスできる場合にアクティブ/アクティブ処理が可能になります。ただし、シングルディスクサイトがネットワークから分離されている場合、そのサイトは削除されます。

RACネットワークハートビート

Oracle RACネットワークハートビートは、クラスタインターコネクト経由でノードに到達できるかどうかを監視します。クラスタに残すには、あるノードが他のノードの半数以上にアクセスできる必要があります。この要件により、2サイトアーキテクチャのRACノード数は次のように選択されます。

- サイトごとに同じ数のノードを配置すると、ネットワーク接続が失われた場合に1つのサイトが削除されます。
- 一方のサイトにN個のノードを配置し、もう一方のサイトにN+1個のノードを配置すると、サイト間接続が失われてネットワーククォーラムに残っているノードの数が多くなり、削除するノードの数が少なくなります。

Oracle 12cR2より前のバージョンでは、サイト障害時にどの側で削除するかを制御することは不可能でした。各サイトのノード数が同じ場合、削除はマスターノード（通常は最初にブートするRACノード）によって制御されます。

Oracle 12cR2では、ノードの重み付け機能が導入されています。この機能により、管理者はOracleによるスプリットブレイン状態の解決方法をより細かく制御できます。簡単な例として、次のコマンドはRAC内の特定のノードの優先順位を設定します。

```
[root@host-a ~]# /grid/bin/crsctl set server css_critical yes
CRS-4416: Server attribute 'CSS_CRITICAL' successfully changed. Restart
Oracle High Availability Services for new value to take effect.
```

Oracle High-Availability Servicesを再起動すると、構成は次のようになります。

```
[root@host-a lib]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
```

ノード `host-a` が重要なサーバとして指定されました。2つのRACノードが分離されている場合は、 `host-a` 生き残って `host-b` 削除されます。

詳細については、Oracleのホワイトペーパー『Oracle Clusterware 12c Release 2 Technical Overview』を参照してください。」

12cR2より前のバージョンのOracle RACでは、CRSログを確認することでマスターノードを特定できます。

```
[root@host-a ~]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
 [root@host-a ~]# grep -i 'master node' /grid/diag/crs/host-
a/crs/trace/crsd.trc
2017-05-04 04:46:12.261525 :   CRSSE:2130671360: {1:16377:2} Master Change
Event; New Master Node ID:1 This Node's ID:1
2017-05-04 05:01:24.979716 :   CRSSE:2031576832: {1:13237:2} Master Change
Event; New Master Node ID:2 This Node's ID:1
2017-05-04 05:11:22.995707 :   CRSSE:2031576832: {1:13237:221} Master
Change Event; New Master Node ID:1 This Node's ID:1
2017-05-04 05:28:25.797860 :   CRSSE:3336529664: {1:8557:2} Master Change
Event; New Master Node ID:2 This Node's ID:1
```

このログは、マスターノードが 2 ノード host-a ID： 1。これはつまり host-a はマスターノードではありません。マスターノードのIDは、コマンドで確認できます。 olsnodes -n。

```
[root@host-a ~]# /grid/bin/olsnodes -n
host-a  1
host-b  2
```

IDがのノード 2 はです host-b`をクリックします。これはマスターノードです。各サイトに同じ数のノードがある構成では、 `host-b 2つのセットが何らかの理由でネットワーク接続を失った場合に存続するサイトです。

マスターノードを識別するログエントリがシステムから期限切れになる可能性があります。この場合、Oracle Cluster Registry（OCR）バックアップのタイムスタンプを使用できます。

```
[root@host-a ~]#  /grid/bin/ocrconfig -showbackup
host-b     2017/05/05 05:39:53     /grid/cdata/host-cluster/backup00.ocr
0
host-b     2017/05/05 01:39:53     /grid/cdata/host-cluster/backup01.ocr
0
host-b     2017/05/04 21:39:52     /grid/cdata/host-cluster/backup02.ocr
0
host-a     2017/05/04 02:05:36     /grid/cdata/host-cluster/day.ocr     0
host-a     2017/04/22 02:05:17     /grid/cdata/host-cluster/week.ocr     0
```

次の例では、マスターノードが host-b。また、マスターノードの変更も示します。 host-a 終了： host-b 5月4日の2時5分から21時39分までの間。マスターノードを識別するこの方法は、前回のOCRバックアップ以降にマスターノードが変更されている可能性があるため、CRSログもチェックされている場合にのみ使用で

きます。この変更が発生した場合は、OCRログに表示されます。

ほとんどのお客様は、環境全体と各サイトで同数のRACノードにサービスを提供する投票ディスクグループを1つ選択しています。ディスクグループは、データベースが格納されているサイトに配置する必要があります。接続が失われると、リモートサイトが削除されます。リモートサイトにはクォーラムがなくなり、データベースファイルにもアクセスできなくなりますが、ローカルサイトは通常どおり稼働し続けます。接続が回復したら、リモートインスタンスを再びオンラインにすることができます。

災害が発生した場合は、サバイバーサイトでデータベースファイルと投票ディスクグループをオンラインにするためにスイッチオーバーが必要です。災害によってAUSOでスイッチオーバーがトリガーされた場合、クラスタが同期されていてストレージリソースが正常にオンラインになるため、NVFAILはトリガーされません。AUSOは非常に高速な操作であり、 `disktimeout` 有効期限が切れます。

サイトが2つしかないため、自動化された外部タイブレークソフトウェアを使用することは不可能であり、強制スイッチオーバーは手動で行う必要があります。

3サイト構成

3つのサイトで拡張RACクラスタを構築する方がはるかに簡単です。MetroClusterシステムの各半分をホストする2つのサイトもデータベースワークロードをサポートし、3つ目のサイトはデータベースとMetroClusterシステムの両方のTiebreakerとして機能します。Oracle Tiebreakerの構成は、第3のサイトに投票に使用するASMディスクグループのメンバーを配置するだけで簡単に構成できます。また、RACクラスタに奇数のノードを配置するために、第3のサイトに運用インスタンスを配置することもできます。

> ⓘ 拡張RAC構成でNFSを使用する場合の重要な情報については、「クォーラム障害グループ」に関するOracleのドキュメントを参照してください。要するに、クォーラムリソースをホストする3番目のサイトへの接続が失われても、プライマリOracleサーバまたはOracle RACプロセスが停止しないように、NFSマウントオプションを変更してsoftオプションを含める必要がある場合があります。

SnapMirrorアクティブ同期

概要

SnapMirror Active Syncを使用すると、非常に高可用性のOracleデータベース環境を構築できます。この環境では、2つの異なるストレージクラスタからLUNを使用できます。

SnapMirrorのアクティブな同期では、データの「プライマリ」コピーと「セカンダリ」コピーはありません。各クラスタはデータのローカルコピーから読み取りIOを提供でき、各クラスタはパートナーに書き込みをレプリケートします。その結果、対称IOビヘイビアが作成されます。

これにより、Oracle RACを両方のサイトで運用インスタンスを持つ拡張クラスタとして実行できます。または、RPO=0のアクティブ/パッシブデータベースクラスタを構築して、サイト停止中にシングルインスタンスデータベースをサイト間で移動できます。このプロセスは、PacemakerやVMware HAなどの製品を使用して自動化できます。これらすべてのオプションの基盤となるのは、SnapMirror Active Syncで管理される同期レプリケーションです。

同期レプリケーション

通常の運用では、1つの例外を除いて、SnapMirrorアクティブ同期は常にRPO=0同期レプリカを提供します。データをレプリケートできない場合、ONTAPでは、データのレプリケートという要件が解除され、一方のサイトのLUNがオフラインになる間に、一方のサイトでIOの提供が再開されます。

ストレージハードウェア

他のストレージディザスタリカバリソリューションとは異なり、SnapMirrorアクティブ同期は非対称プラットフォームの柔軟性を提供します。各サイトのハードウェアが同一である必要はありません。この機能を使用すると、SnapMirrorアクティブ同期をサポートするために使用するハードウェアのサイズを適正化できます。リモートストレージシステムは、本番環境のワークロードを完全にサポートする必要がある場合はプライマリサイトと同一にすることができますが、災害によってI/Oが減少した場合は、リモートサイトの小規模システムよりも対費用効果が高くなります。

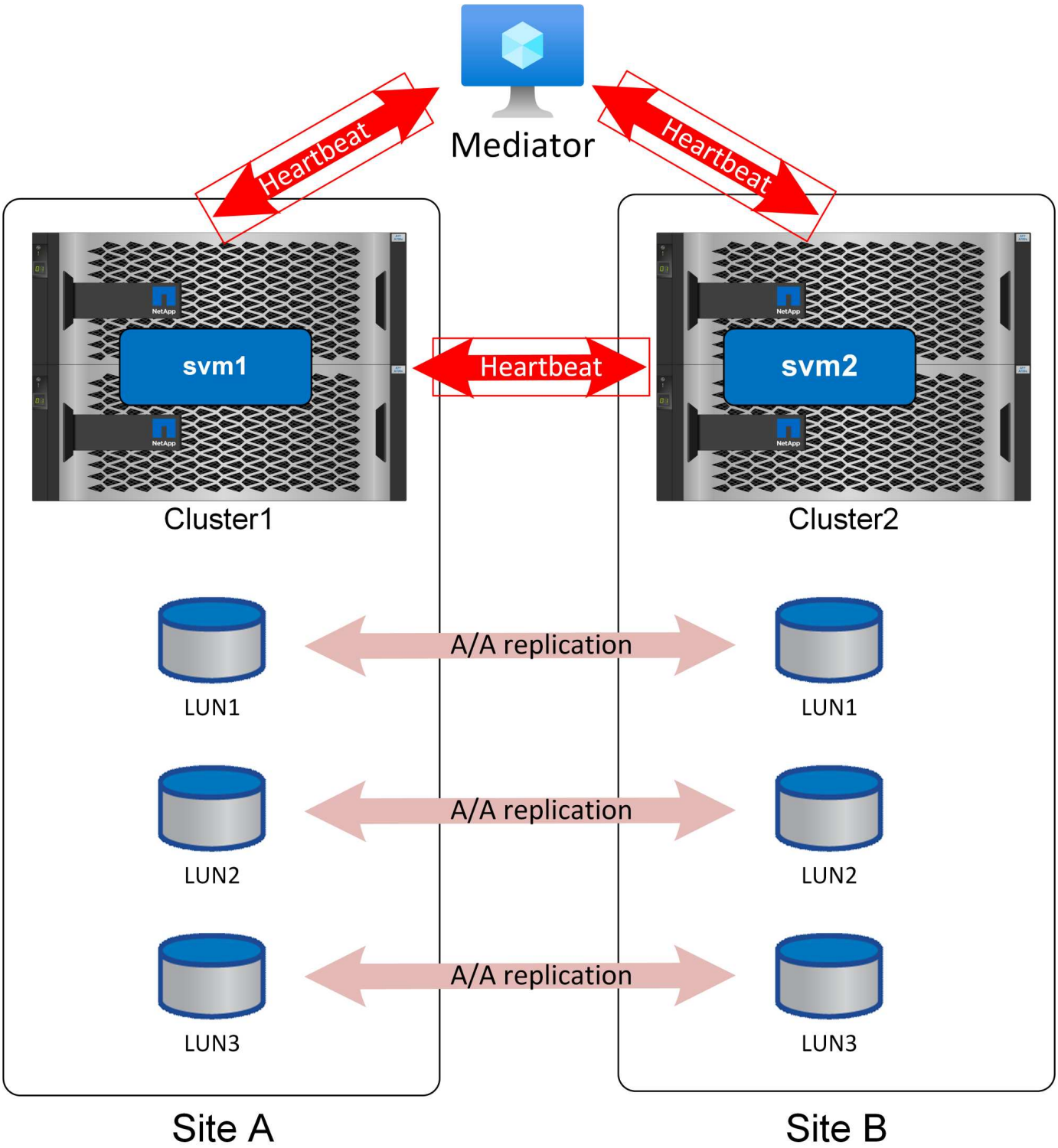
ONTAPメディエーター

ONTAPメディエーターは、NetAppサポートからダウンロードするソフトウェアアプリケーションで、通常は小規模な仮想マシンに導入されます。ONTAPメディエーターは、SnapMirrorのアクティブな同期ではTiebreakerになりません。これは、SnapMirrorのアクティブな同期レプリケーションに含まれる2つのクラスタの代替通信チャネルです。自動処理は、パートナーから直接接続またはメディエーター経由で受け取った応答に基づいてONTAPによって実行されます。

ONTAPメディエーター

フェイルオーバーを安全に自動化するにはメディエーターが必要です。理想的には、独立した3つ目のサイトに配置しますが、レプリケーションに参加しているクラスタの1つと同じ場所に配置すれば、ほとんどのニーズに対応できます。

調停者は実際にはタイブレーカーではありませんが、実質的にはそれがその機能を提供します。メディエーターは、クラスター ノードの状態を判断するのに役立ち、サイト障害が発生した場合の自動切り替えプロセスを支援します。Mediator はいかなる状況でもデータを転送しません。



自動フェイルオーバーの最大の課題はスプリットブレインの問題であり、この問題は2つのサイト間の接続が失われた場合に発生します。何が起こるべきでしょうか? 2つの異なるサイトがデータのサバイバーコピーとして自分自身を指定する必要はありませんが、1つのサイトでは、反対側のサイトが実際に失われたことと、反対側のサイトと通信できないことを区別するにはどうすればよいでしょうか。

ここでメディエーターが写真に入ります3番目のサイトに配置され、各サイトからそのサイトへの個別のネットワーク接続がある場合は、他のサイトの正常性を検証するための追加のパスが各サイトに用意されています。上の図をもう一度見て、次のシナリオを検討してください。

- 一方または両方のサイトからメディエーターに障害が発生した場合、またはメディエーターに到達できない場合はどうなりますか？
  - 2つのクラスタは、レプリケーションサービスに使用されるのと同じリンクを介して相互に通信できます。
  - データは引き続きRPO=0の保護で提供される
- サイトAに障害が発生した場合の動作
  - サイトBは、両方の通信チャネルがダウンしたことを確認します。
  - サイトBがデータサービスをテイクオーバーするが、RPO=0ミラーリングなし
- サイトBで障害が発生した場合の動作
  - サイトAでは、両方の通信チャネルがダウンしていることが確認されます。
  - サイトAがデータサービスをテイクオーバーするが、RPO=0ミラーリングなし

もう1つ考慮すべきシナリオがあります。データレプリケーションリンクの停止です。サイト間のレプリケーションリンクが失われた場合、RPO=0のミラーリングは明らかに不可能です。ではどうすればいいのでしょうか。

これは、優先サイトのステータスによって制御されます。SM-AS関係では、一方のサイトがもう一方のサイトのセカンダリになります。これは通常の運用には影響せず、すべてのデータアクセスは対称的ですが、レプリケーションが中断された場合は、運用を再開するためにこの関係を解除する必要があります。その結果、優先サイトはミラーリングなしで処理を継続し、レプリケーション通信がリストアされるまでセカンダリサイトはIO処理を停止します。

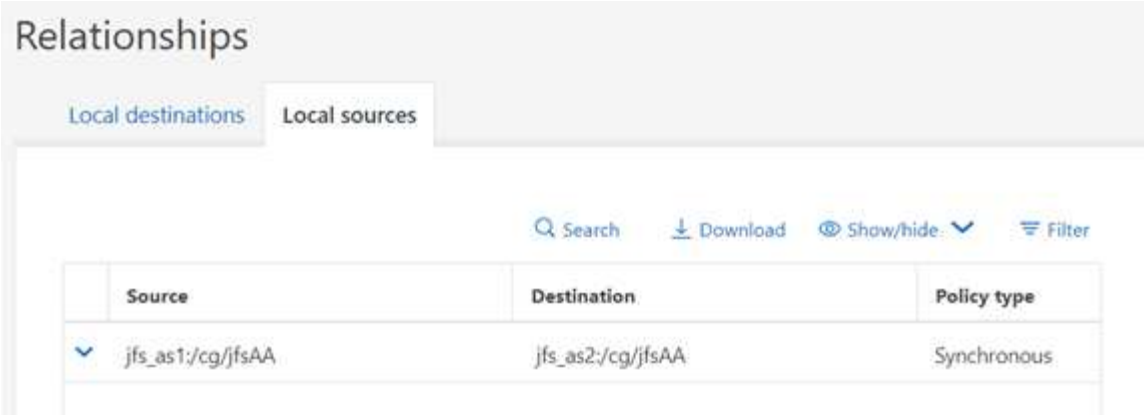
SnapMirrorアクティブ同期優先サイト

SnapMirrorのアクティブな同期の動作は対称ですが、重要な例外が1つあります（推奨サイト構成）。

SnapMirrorアクティブ同期では、一方のサイトが「ソース」で、もう一方が「デスティネーション」と見なされます。これは一方向のレプリケーション関係を意味しますが、IO動作には適用されません。レプリケーションは双方向であり、対称であり、IO応答時間はミラーの両側で同じです。

> `source`指定は、優先サイトを制御します。レプリケーションリンクが失われた場合、ソースコピー上のLUNパスは引き続きデータを提供しますが、デスティネーションコピー上のLUNパスは、レプリケーションが再確立されてSnapMirrorが同期状態に戻るまで使用できなくなります。その後、パスでデータの提供が再開されます。

ソース/デスティネーションの設定はSystemManagerで確認できます。



または、CLIで次の操作を行います。

```
Cluster2::> snapmirror show -destination-path jfs_as2:/cg/jfsAA

                           Source Path: jfs_as1:/cg/jfsAA
                      Destination Path: jfs_as2:/cg/jfsAA
                     Relationship Type: XDP
                Relationship Group Type: consistencygroup
                    SnapMirror Schedule: -
                 SnapMirror Policy Type: automated-failover-duplex
                      SnapMirror Policy: AutomatedFailOverDuplex
                            Tries Limit: -
                      Throttle (KB/sec): -
                           Mirror State: Snapmirrored
                    Relationship Status: InSync
```

重要なのは、ソースがcluster1のSVMであることです。前述のように、「ソース」と「デスティネーション」という用語は、レプリケートされたデータのフローを表していません。両方のサイトが書き込みを処理し、反対側のサイトにレプリケートできます。実際には、両方のクラスタがソースとデスティネーションです。1つのクラスタをソースとして指定すると、レプリケーションリンクが失われた場合に、どのクラスタが読み取り/書き込みストレージシステムとして残っているかが制御されます。

ネットワークトポロジ

均一なアクセス

統一されたアクセスネットワークとは、ホストが両方のサイト（または同じサイト内の障害ドメイン）のパスにアクセスできることを意味します。

SM-ASの重要な機能の1つは、ホストがどこにあるかを認識するようにストレージシステムを設定できることです。LUNを特定のホストにマッピングするときに、LUNが特定のストレージシステムに近接しているかどうかを指定できます。

近接設定

プロキシミティとは、特定のホストWWNまたはiSCSIイニシエータIDがローカルホストに属していることを

示すクラスタ単位の構成を指します。これは、LUNアクセスを設定するための2番目のオプションの手順です。

最初の手順では、通常のigroup設定を行います。各LUNは、そのLUNにアクセスする必要があるホストのWWN/iSCSI IDを含むigroupにマッピングする必要があります。これは、どのホストがLUNに_access_toを持つかを制御します。

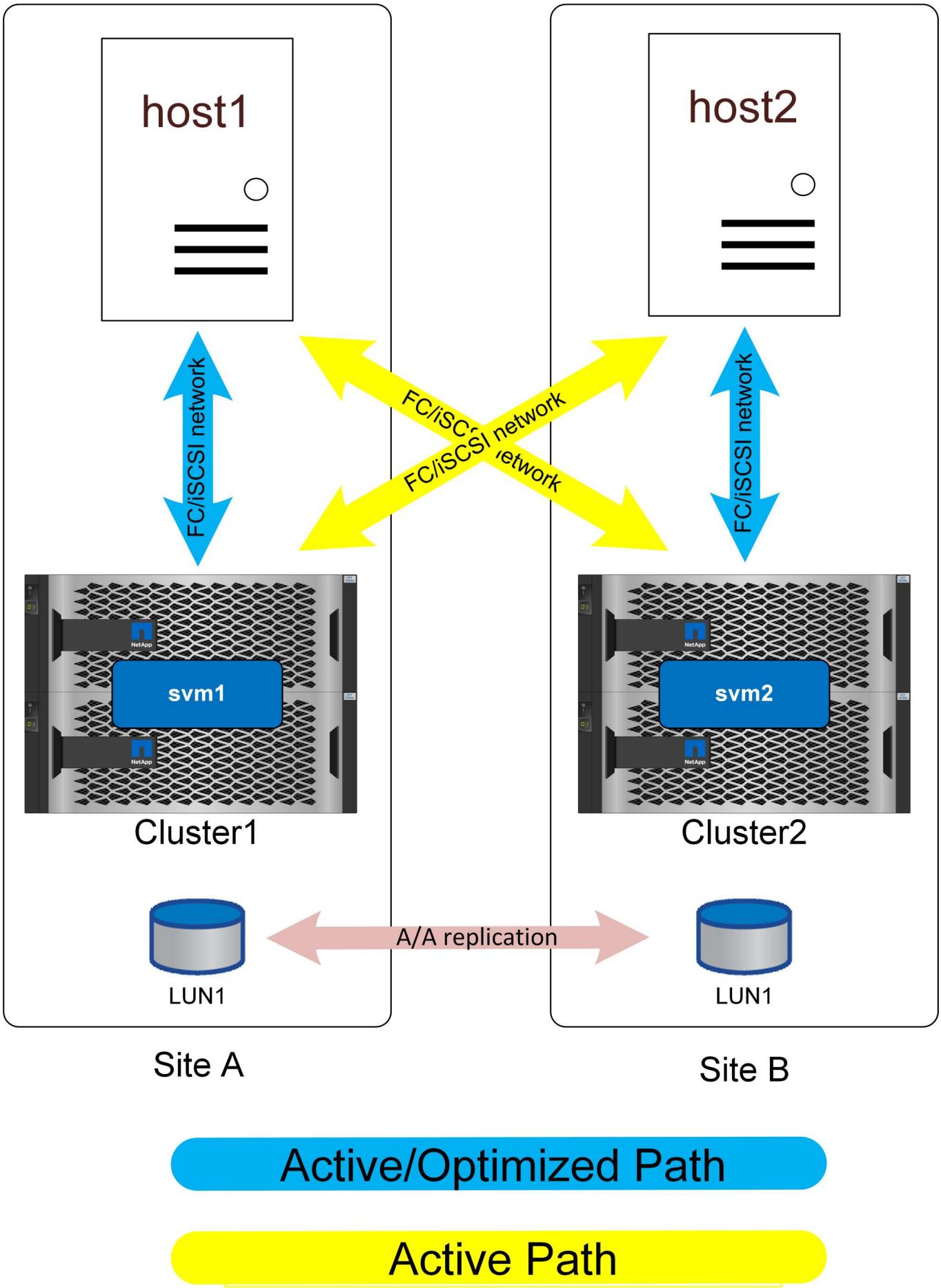
2番目のオプション手順は、ホストプロキシミティを設定することです。これはアクセスを制御するのではなく、_priority_を制御します。

たとえば、サイトAのホストがSnapMirror Active Syncで保護されているLUNにアクセスするように設定されている場合、SANがサイト間で拡張されるため、サイトAのストレージまたはサイトBのストレージを使用してそのLUNへのパスを使用できます。

近接設定を使用しない場合、両方のストレージシステムがアクティブな最適パスをアドバタイズするため、そのホストは両方のストレージシステムを均等に使用します。SANのレイテンシやサイト間の帯域幅に制限がある場合は、この設定を解除できない可能性があります。また、通常動作中に各ホストがローカルストレージシステムへのパスを優先的に使用するように設定することもできます。これは、ホストWWN/iSCSI IDをローカルクラスタに近接ホストとして追加することで設定します。これは、CLIまたはSystemManagerで実行できます。

AFF

AFFシステムでホストプロキシミティが設定されている場合、パスは次のように表示されます。


host1
host2
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
svm1
svm2
Cluster1
Cluster2
A/A replication
LUN1
LUN1
Site A
Site B
Active/Optimized Path
Active Path

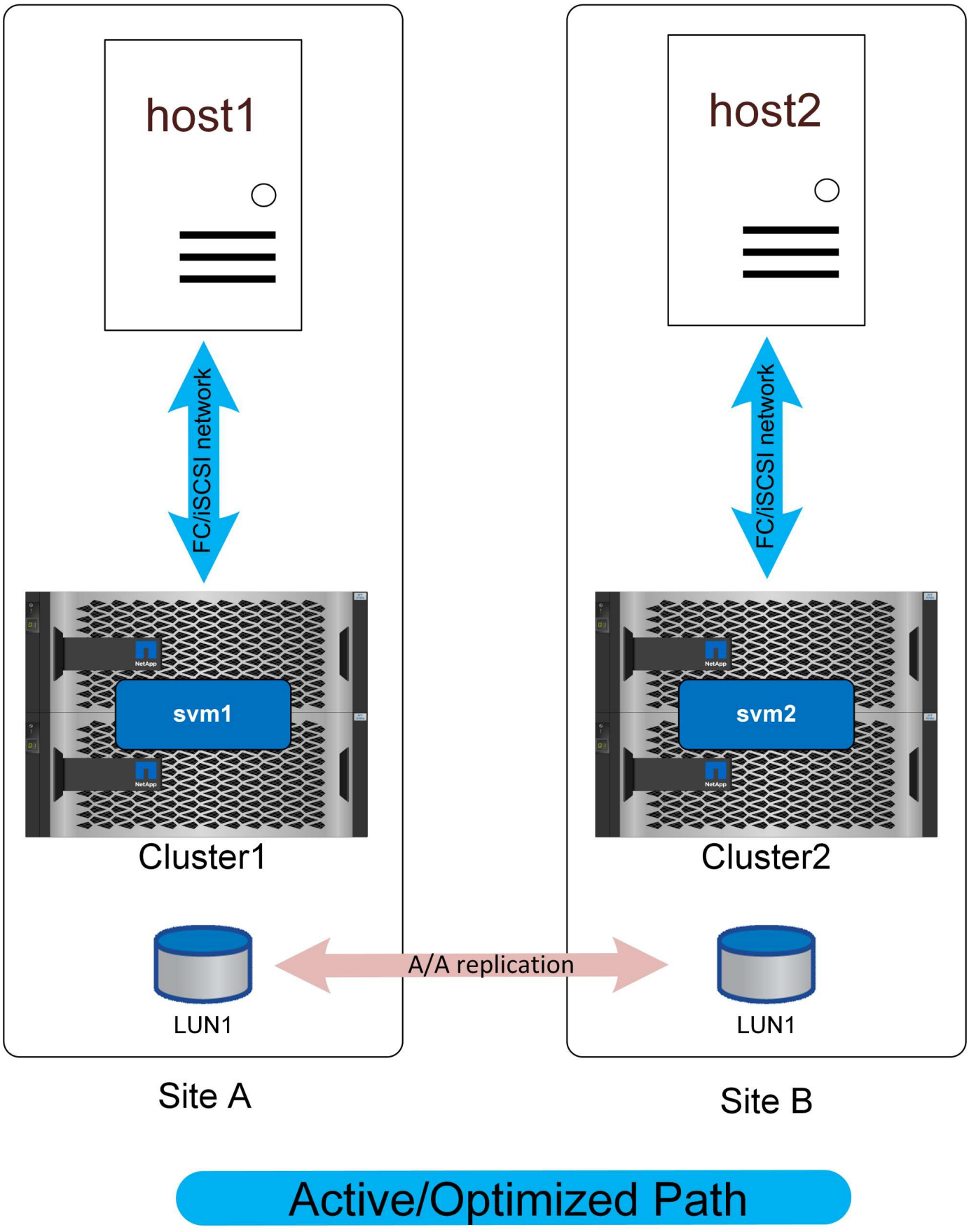
通常の運用では、すべてのIOがローカルIOになります。読み取りと書き込みはローカルストレージアレイから処理されます。もちろん、書き込みIOも確認応答の前にローカルコントローラでリモートシステムにレプリケートする必要がありますが、すべての読み取りIOはローカルで処理されるため、サイト間のSANリンクを経由して余分なレイテンシが発生することはありません。

非最適パスが使用されるのは、すべてのアクティブ/最適パスが失われた場合だけです。たとえば、サイトAのアレイ全体に電力が供給されなくなっても、サイトAのホストはサイトBのアレイへのパスに引き続きアクセスできるため、レイテンシは高くなりますが運用を継続できます。

この図では、わかりやすいように、ローカルクラスタを経由する冗長パスを示していません。ONTAPストレージシステム自体はHAであるため、コントローラ障害が発生してもサイト障害は発生しません。影響を受けるサイトで使用されるローカルパスが変更されるだけです。

ASA

NetApp ASAシステムは、クラスタ上のすべてのパスでアクティブ/アクティブマルチパスを提供します。これはSM-AS設定にも適用されます。



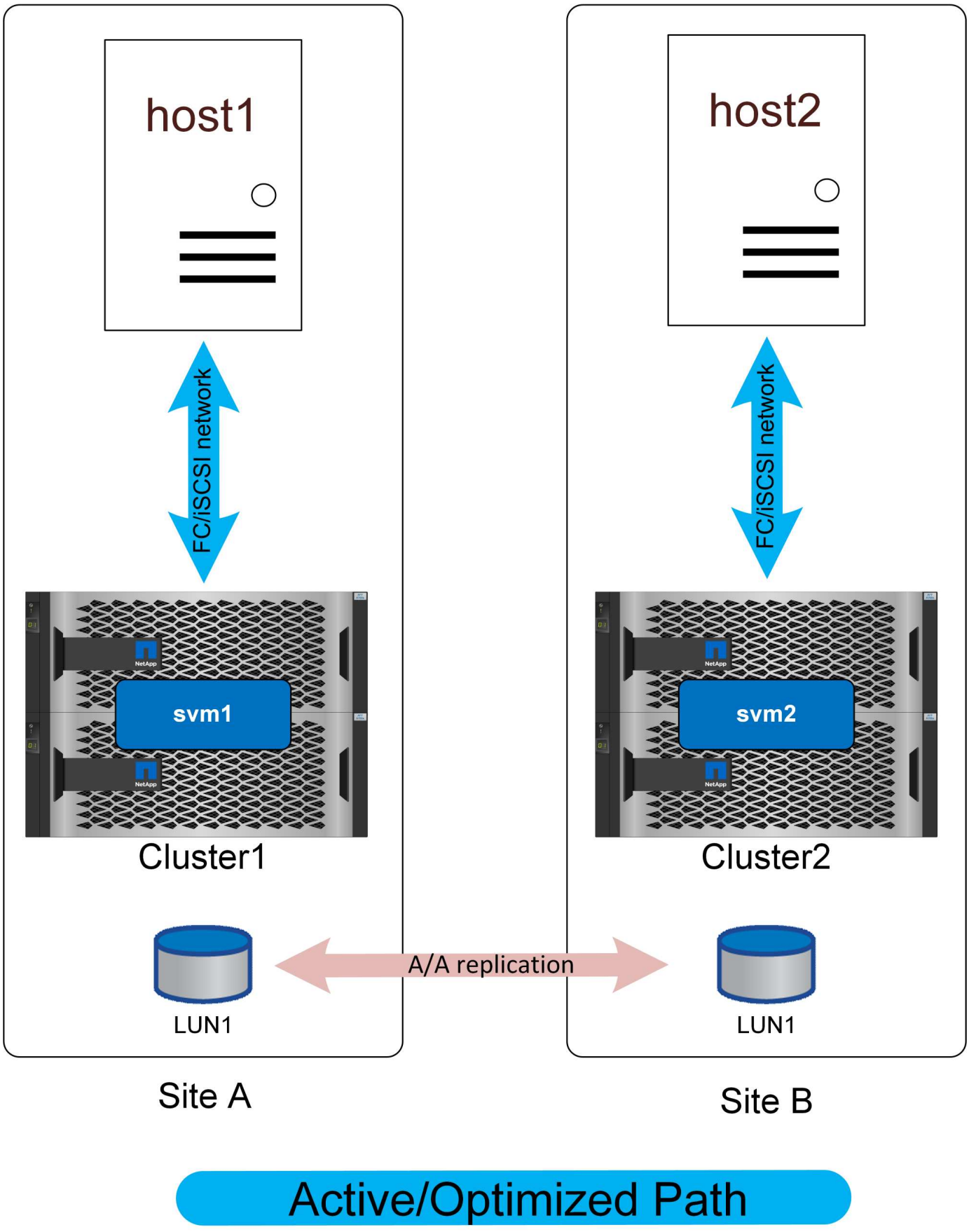
アクセスが不均一なASA構成は、AFFの場合とほとんど同じように機能します。アクセスが統一されている場合、IOはWANを通過します。これは望ましい場合とそうでない場合があります。

2つのサイトがファイバ接続で100m離れている場合、WANを経由する追加のレイテンシは検出されませんが、サイト間の距離が離れていると、両方のサイトで読み取りパフォーマンスが低下します。対照的に、AFFでは、これらのWAN交差パスは使用可能なローカルパスがない場合にのみ使用され、すべてのIOがローカルIOになるため、日 々 のパフォーマンスが向上します。不均一なアクセスネットワークを使用するASAは、サイト間の遅延アクセスペナルティを発生させることなく、ASAのコストと機能のメリットを得るためのオプションです。

低レイテンシ構成でSM-ASを使用するASAには、2つの興味深い利点があります。まず、I/Oは2倍のパスを使用して2倍のコントローラで処理できるため、1台のホストのパフォーマンスが実質的に2倍になります。2つ目は、単一サイト環境では、ホストへのアクセスを中断することなくストレージシステム全体が失われる可能性があるため、非常に高い可用性を提供することです。

不均一なアクセス

非ユニフォームアクセスネットワークとは、各ホストがローカルストレージシステム上のポートにしかアクセスできないことを意味します。SANを複数のサイト（または同じサイト内の障害ドメイン）に拡張することはできません。



このアプローチの主なメリットはSANのシンプルさです。SANをネットワーク経由で拡張する必要がなくなります。お客様によっては、サイト間の接続遅延が十分でない場合や、サイト間ネットワーク経由でFC SAN

トラフィックをトンネリングするためのインフラストラクチャが不足している場合があります。

不均一なアクセスの欠点は、レプリケーションリンクの喪失などの特定の障害シナリオで、一部のホストがストレージにアクセスできなくなることです。ローカルストレージの接続が失われると、単一のホストでのみ実行されている非クラスタデータベースなど、単一インスタンスとして実行されるアプリケーションは失敗します。データは保護されますが'データベース・サーバはアクセスできなくなりますリモートサイトで、できれば自動化されたプロセスを使用して再起動する必要があります。たとえば、VMware HAは、あるサーバでオールパスダウン状態を検出し、パスが使用可能な別のサーバでVMを再起動できます。

一方、Oracle RACなどのクラスタ化されたアプリケーションは、2つの異なるサイトで同時に利用可能なサービスを提供できます。サイトが失われても、アプリケーションサービス全体が失われるわけではありません。サバイバーサイトでは、引き続きインスタンスを使用して実行できます。

多くの場合、サイト間リンク経由でストレージにアクセスするアプリケーションによるレイテンシのオーバーヘッドは許容できません。つまり、サイトのストレージが失われると、障害が発生したサイトのサービスをシャットダウンする必要が生じるため、統一されたネットワークの可用性の向上は最小限で済みます。

> この図では、わかりやすいように、ローカルクラスタを経由する冗長パスを示していません。ONTAPストレージシステム自体はHAであるため、コントローラ障害が発生してもサイト障害は発生しません。影響を受けるサイトで使用されるローカルパスが変更されるだけです。

Oracleの構成

概要

SnapMirrorアクティブ同期を使用しても、データベースの運用に関するベストプラクティスが追加または変更されるとは限りません。

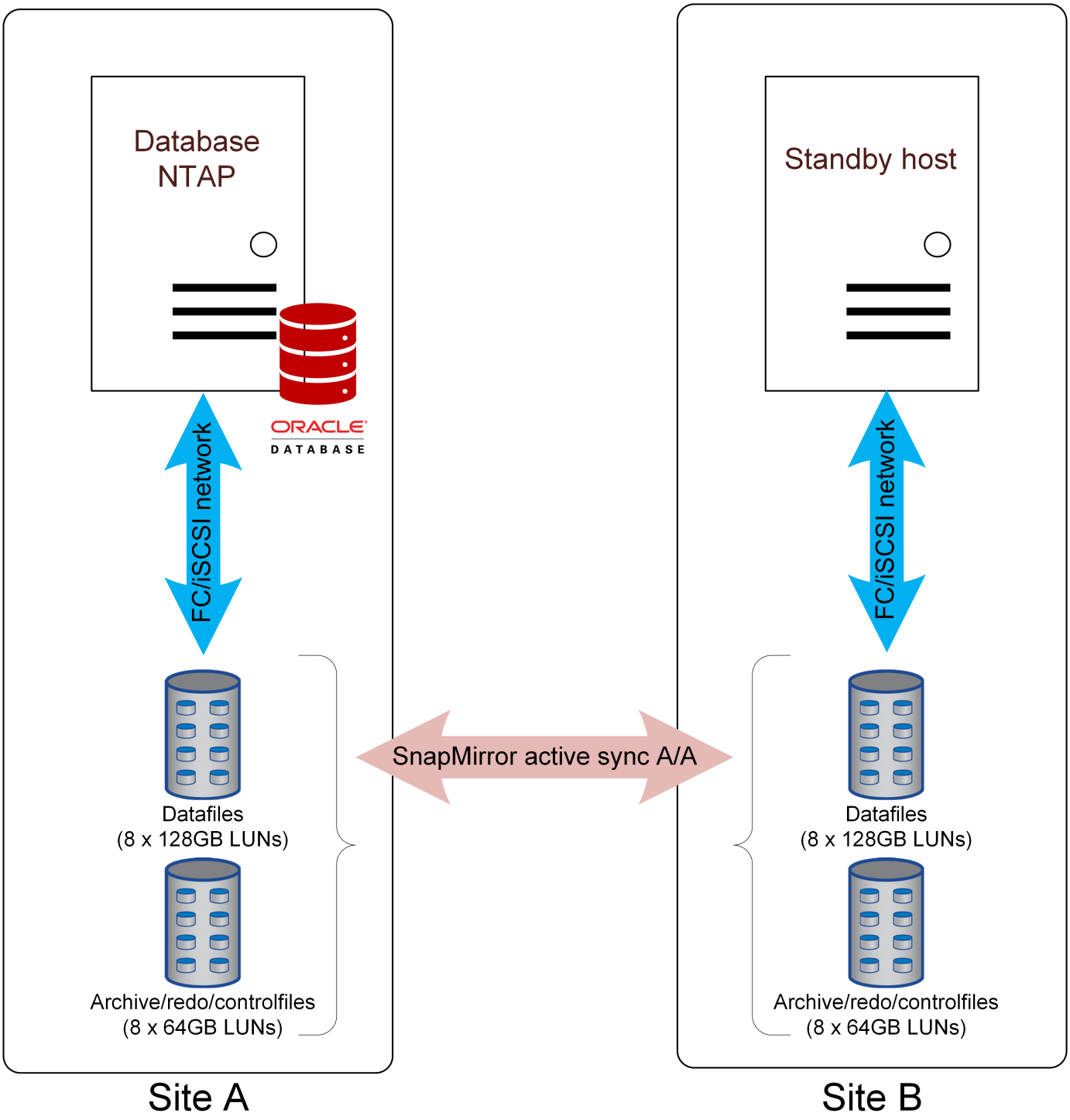
最適なアーキテクチャは、ビジネス要件によって異なります。たとえば、データ損失に対するRPO=0の保護が目標であるにもかかわらず、RTOが緩和されている場合は、Oracleのシングルインスタンスデータベースを使用し、SM-ASでLUNをレプリケートすれば十分であり、Oracleのライセンスの問題からより安価になる可能性があります。リモートサイトに障害が発生しても運用は中断されず、プライマリサイトが停止すると、サバイバーサイトのLUNはオンラインで使用可能な状態になります。

RTOの方が厳しい場合は、スクリプトやPacemakerやAnsibleなどのクラスタウェアを使用した基本的なアクティブ/パッシブ自動化を使用すると、フェイルオーバー時間が短縮されます。たとえば、プライマリサイトでVMの障害を検出し、リモートサイトでVMをアクティブにするようにVMware HAを設定できます。

最後に、きわめて高速なフェイルオーバーを実現するために、Oracle RACを複数のサイトに導入できます。データベースは常にオンラインで両方のサイトで利用できるため、RTOは基本的にゼロになります。

Oracleシングルインスタンス

以下に説明する例は、SnapMirrorアクティブ同期レプリケーションを使用してOracleシングルインスタンスデータベースを導入するための多数のオプションの一部を示しています。



事前設定された**OS**を使用したフェイルオーバー

SnapMirror Active Syncはディザスタリカバリサイトにデータの同期コピーを作成しますが、そのデータを利用できるようにするには、オペレーティングシステムと関連するアプリケーションが必要です。基本的な自動化により、環境全体のフェイルオーバー時間を大幅に短縮できます。PacemakerなどのClusterware製品は、サイト間でクラスタを作成するためによく使用されます。多くの場合、フェイルオーバープロセスは単純なスクリプトで実行できます。

プライマリノードが失われると、クラスタウェア（またはスクリプト）によって代替サイトでデータベースがオンラインになります。1つは、データベースを構成するSANリソース用に事前設定されたスタンバイサーバ

を作成する方法です。プライマリサイトに障害が発生すると、クラスタウェアまたはスクリプト化された代替サイトが次のような一連の処理を実行します。

1. プライマリサイトの障害を検出
2. FCまたはiSCSI LUNの検出の実行
3. ファイルシステムのマウント、ASMディスクグループのマウント
4. データベースの起動

このアプローチの主な要件は、リモートサイトでOSを実行することです。Oracleバイナリを使用して事前に設定する必要があります。つまり、Oracleのパッチ適用などのタスクをプライマリサイトとスタンバイサイトで実行する必要があります。また、災害が発生した場合は、Oracleバイナリをリモートサイトにミラーリングしてマウントすることもできます。

実際のアクティベーション手順は簡単です。LUN検出などのコマンドでは、FCポートあたりのコマンド数が少なくて済みます。ファイルシステムのマウントはコマンドにすぎませ `mount`ん。データベースとASMの両方を、1つのコマンドでCLIから開始および停止できます。

仮想**OS**によるフェイルオーバー

データベース環境のフェイルオーバーを拡張して、オペレーティングシステム自体を含めることができます。理論的には、このフェイルオーバーはブートLUNで実行できますが、ほとんどの場合、仮想OSで実行されます。手順の手順は次のようになります。

1. プライマリサイトの障害を検出
2. データベースサーバ仮想マシンをホストするデータストアのマウント
3. 仮想マシンの起動
4. データベースを手動で起動するか、仮想マシンでデータベースが自動的に起動するように設定します。

たとえば、ESXクラスタが複数のサイトにまたがっているとします。災害が発生した場合は、スイッチオーバー後にディザスタリカバリサイトで仮想マシンをオンラインにすることができます。

ストレージ障害からの保護

上の図は"不均一なアクセス"、の使用方法を示しています。SANが複数のサイトにまたがっているわけではありません。これは設定が簡単で、現在のSAN機能では唯一の選択肢となる場合もありますが、プライマリストレージシステムに障害が発生すると、アプリケーションがフェイルオーバーされるまでデータベースが停止します。

耐障害性を高めるために、このソリューションをとともに導入することもでき"均一なアクセス"ます。これにより、アプリケーションは、反対側のサイトからアドバタイズされたパスを使用して動作を継続できます。

Oracle拡張**RAC**

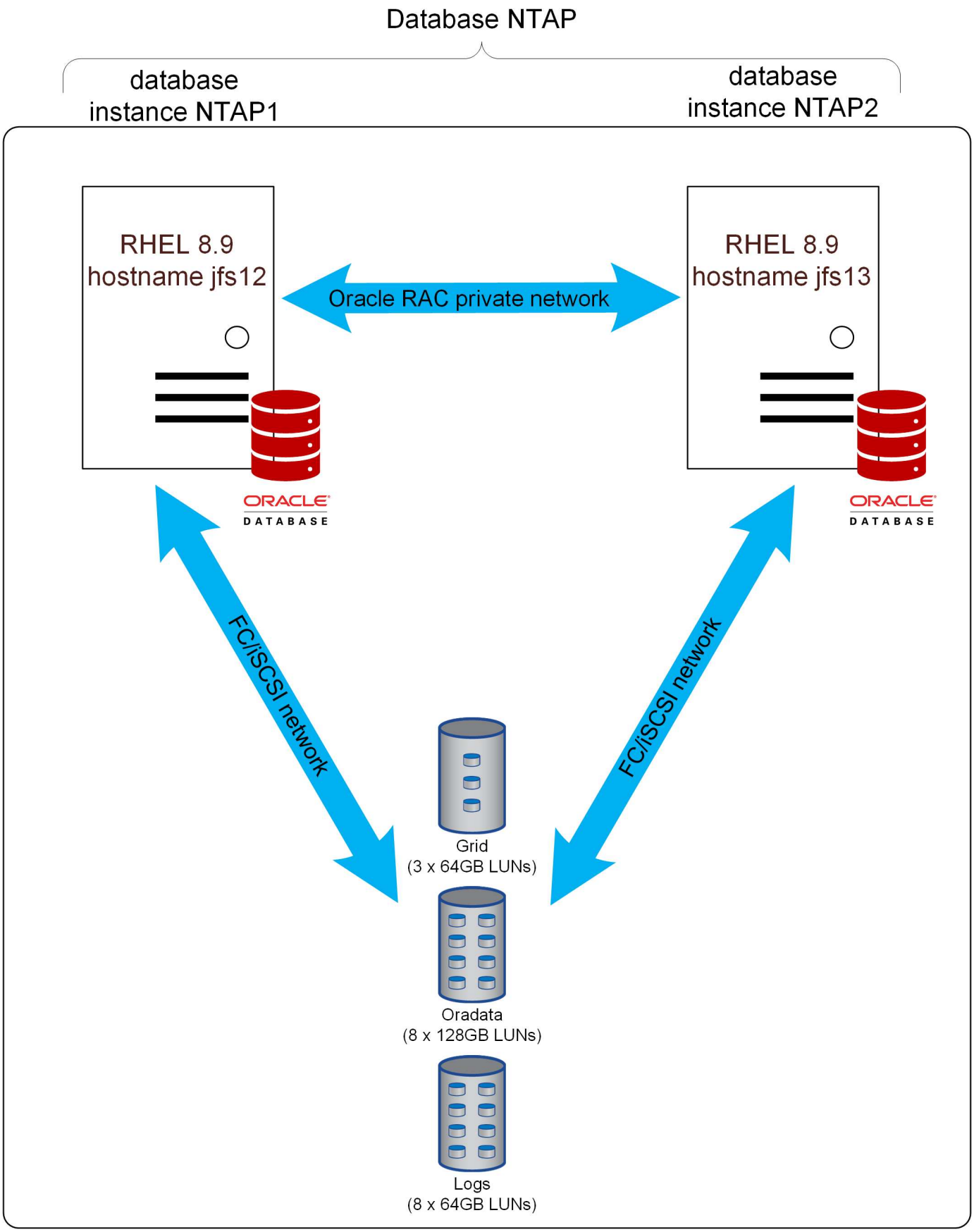
多くのお客様が、Oracle RACクラスタを複数のサイトにまたがって構成し、完全なアクティブ/アクティブ構成を実現することで、RTOを最適化しています。Oracle RACのクォーラム管理を含める必要があるため、設計全体が複雑になります。

従来の拡張RACクラスタでは、ASMミラーリングを使用してデータを保護していました。このアプローチは機能しますが、多くの手動設定手順が必要になり、ネットワークインフラストラクチャにオーバーヘッドが発

生します。一方、SnapMirrorのアクティブな同期機能でデータレプリケーションを実行できるようにすることで、ソリューションが大幅に簡易化されます。同期、中断後の再同期、フェイルオーバー、クォーラム管理などの操作が容易になります。また、SANの設計と管理を簡素化するために、SANをサイト間に分散させる必要もありません。

レプリケーション

SnapMirrorアクティブ同期のRAC機能を理解するには、ストレージをミラーリングされたストレージでホストされている単一のLUNセットとして表示することが重要です。例：



プライマリコピーまたはミラーコピーはありません。論理的には、各LUNのコピーは1つだけで、そのLUNは2つの異なるストレージシステム上にあるSANパスで使用できます。ホストから見ると、ストレージフェイルオーバーは発生せず、代わりにパスが変更されます。さまざまな障害イベントが発生すると、LUNへの特定

のパスが失われても、他のパスはオンラインのままになる可能性があります。SnapMirrorのアクティブな同期により、すべての運用パスで同じデータを利用できるようになります。

ストレージ構成

この構成例では、ASMディスクは、エンタープライズストレージの単一サイトRAC構成と同じように設定されています。ストレージシステムはデータ保護を提供するため、ASM外部冗長性が使用されます。

ユニフォームアクセスと非インフォームアクセス

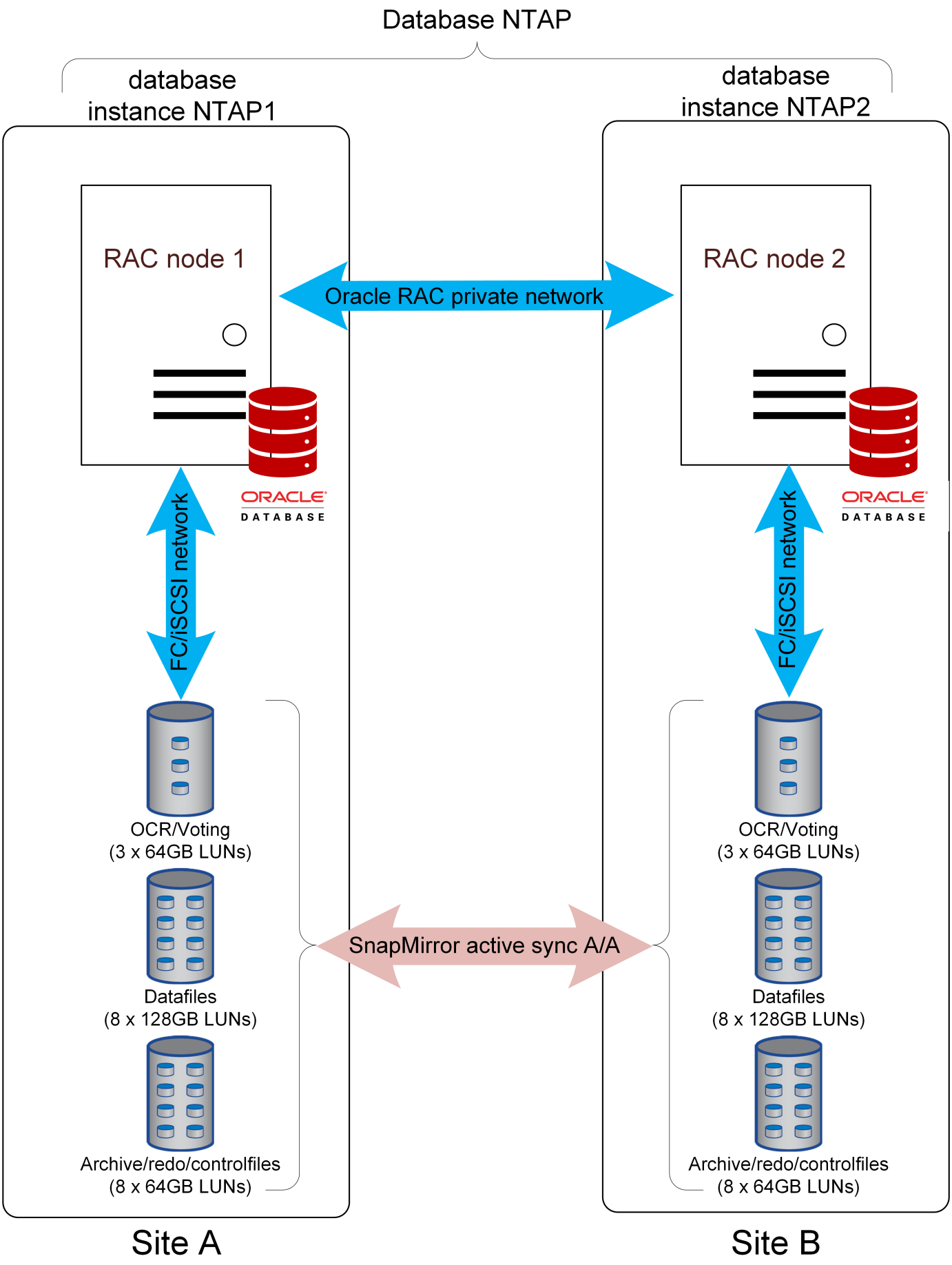
SnapMirrorアクティブ同期上のOracle RACで最も重要な考慮事項は、均一アクセスと非均一アクセスのどちらを使用するかです。

アクセスが統一されているため、各ホストは両方のクラスタのパスを認識できます。一様でないアクセスとは、ホストがローカルクラスタへのパスのみを認識できることを意味します。

どちらのオプションも特に推奨または推奨されないものではありません。ダークファイバを使用してサイトを接続しているお客様もいれば、そのような接続を利用していないお客様や、SANインフラで長距離ISLをサポートしていないお客様もいます。

不均一なアクセス

アクセスが一様でない場合、SANの観点からはより簡単に設定できます。


Database NTAP
database instance NTAP1
database instance NTAP2
RAC node 1
RAC node 2
Oracle RAC private network
FC/iSCSI network
FC/iSCSI network
ORACLE DATABASE
ORACLE DATABASE
OCR/Voting (3 x 64GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
SnapMirror active sync A/A
OCR/Voting (3 x 64GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Site A
Site B

このアプローチの主な欠点"不均一なアクセス"は、サイト間のONTAP接続が失われたり、ストレージシステムが失われたりすると、一方のサイトのデータベースインスタンスが失われることです。これは明らかに望ましくありませんが、シンプルなSAN構成と引き換えに許容可能なリスクになる可能性があります。

均一なアクセス

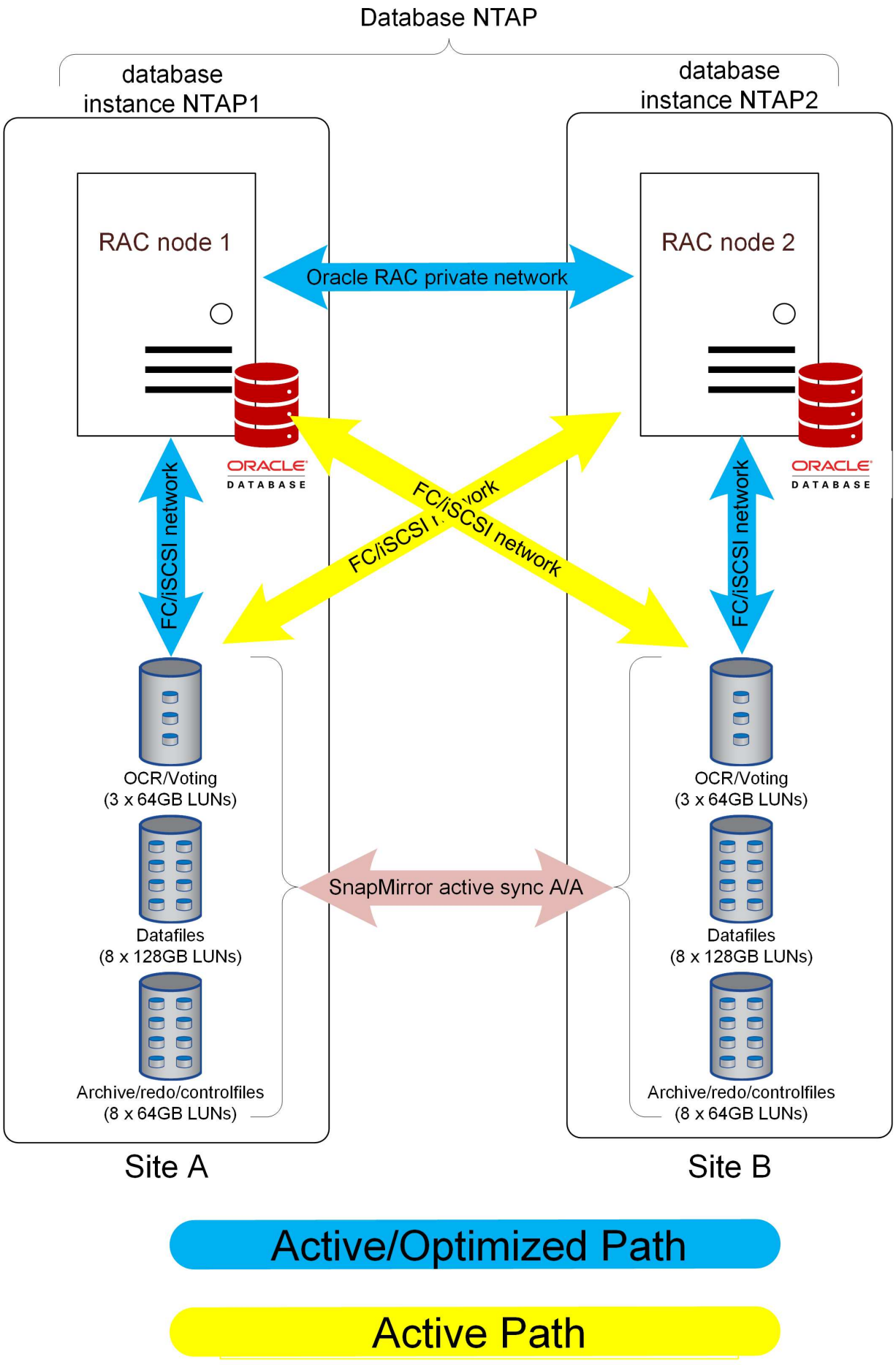
アクセスを統一するには、SANをサイト間に拡張する必要があります。主なメリットは、ストレージシステムが停止してもデータベースインスタンスが失われないことです。その結果、現在使用されているパスがマルチパスに変更されます。

不均一アクセスを設定するには、いくつかの方法があります。

次の図には、単純なコントローラ障害時に使用されるアクティブだが最適化されていないパスもありますが、この図では省略しています。

近接設定を使用した**AFF**

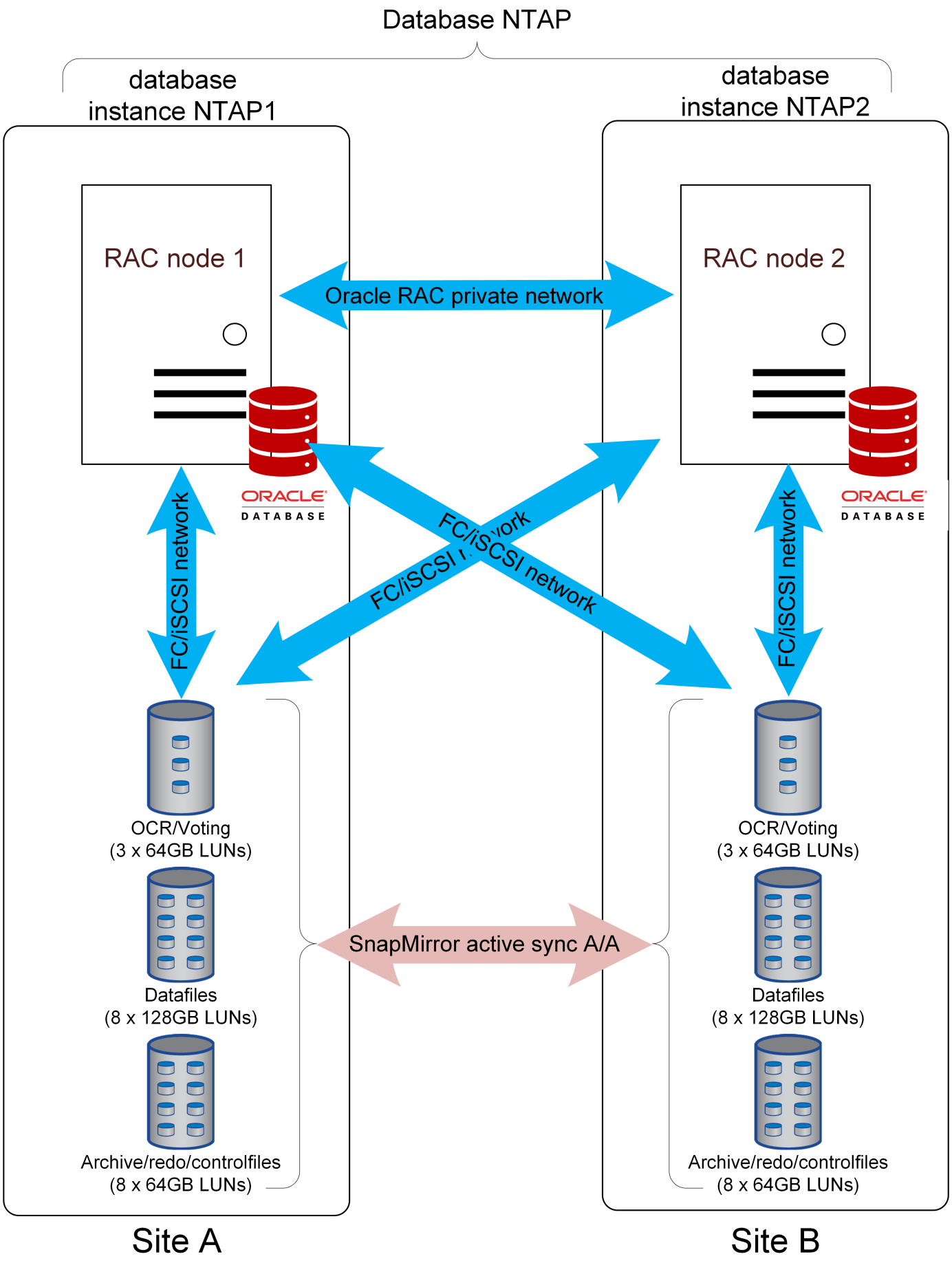
サイト間のレイテンシが大きい場合は、ホストとの近接設定を使用してAFFシステムを設定できます。これにより、各ストレージシステムはどのホストがローカルでどのホストがリモートであるかを認識し、パスの優先順位を適切に割り当てることができます。


Database NTAP
database instance NTAP1
database instance NTAP2
RAC node 1
RAC node 2
Oracle RAC private network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
ORACLE DATABASE
ORACLE DATABASE
OCR/Voting (3 x 64GB LUNs)
OCR/Voting (3 x 64GB LUNs)
SnapMirror active sync A/A
Datafiles (8 x 128GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Site A
Site B
Active/Optimized Path
Active Path

通常運用時は、各Oracleインスタンスがローカルのアクティブ/最適パスを優先的に使用します。その結果、すべての読み取りはブロックのローカルコピーによって処理されます。これにより、レイテンシが最小限に抑えられます。書き込みIOも同様に、ローカルコントローラへのパスに送信されます。IOは確認される前にレプリケートする必要がありますが、その場合もサイト間ネットワークを通過するための追加のレイテンシが発生しますが、同期レプリケーションソリューションではこれを回避することはできません。

近接設定なしの**ASA / AFF**

サイト間のレイテンシがそれほど高くない場合は、ホストとの近接を設定せずにAFFシステムを構成するか、ASAを使用できます。


Database NTAP
database instance NTAP1
database instance NTAP2
RAC node 1
RAC node 2
Oracle RAC private network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
FC/iSCSI network
ORACLE DATABASE
ORACLE DATABASE
OCR/Voting (3 x 64GB LUNs)
OCR/Voting (3 x 64GB LUNs)
SnapMirror active sync A/A
Datafiles (8 x 128GB LUNs)
Datafiles (8 x 128GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Archive/redo/controlfiles (8 x 64GB LUNs)
Site A
Site B

各ホストが両方のストレージシステムのすべての動作パスを使用できるようになります。これにより、各ホストが1つだけでなく2つのクラスタの潜在的なパフォーマンスを利用できるようになるため、パフォーマンスが大幅に向上します。

ASAでは、両方のクラスタへのすべてのパスがアクティブで最適化されているとみなされるだけでなく、パートナーコントローラのパスもアクティブになります。その結果、常にクラスタ全体でオールアクティブなSANパスが作成されます。

ASAシステムは、不均一なアクセス設定でも使用できます。サイト間パスは存在しないため、IOがISLを経由してもパフォーマンスに影響はありません。

RAC Tiebreaker

SnapMirrorアクティブ同期を使用した拡張RACはIOに関して対称アーキテクチャですが、スプリットブレイン管理に接続される例外が1つあります。

レプリケーションリンクが失われ、どちらのサイトにもクォーラムがない場合はどうなりますか。何が起こるべきでしょうか? この質問は、Oracle RACとONTAPの両方の動作に当てはまります。サイト間で変更をレプリケートできない場合に運用を再開するには、一方のサイトを停止し、もう一方のサイトを使用できなくなる必要があります。

は、"ONTAPメディエーター"ONTAPレイヤでこの要件に対応します。RACのタイブレークには複数のオプションがあります。

Oracleタイブレーカー

Oracle RACのスプリットブレインリスクを管理する最善の方法は、奇数のRACノードを使用すること（できれば3つ目のサイトのTiebreakerを使用すること）です。3つ目のサイトが使用できない場合は、Tiebreakerインスタンスを2つのサイトの一方のサイトに配置して、優先サバイバーサイトに指定できます。

Oracleおよび**CSS_CRITICAL**

ノード数が偶数の場合、Oracle RACのデフォルトの動作では、クラスタ内のいずれかのノードの重要度が他のノードよりも高くなります。優先度の高いノードを含むサイトはサイト分離を継続し、もう一方のサイトのノードは削除されます。優先順位付けは複数の要因に基づいて行われますが、設定を使用してこの動作を制御することもできます `css_critical`。

"例"アーキテクチャでは、RACノードのホスト名はjfs12およびjfs13です。の現在の設定は `css_critical`次のとおりです。

```
[root@jfs12 ~]# /grid/bin/crsctl get server css_critical
CRS-5092: Current value of the server attribute CSS_CRITICAL is no.

[root@jfs13 trace]# /grid/bin/crsctl get server css_critical
CRS-5092: Current value of the server attribute CSS_CRITICAL is no.
```

jfs12が設定されたサイトを優先サイトにする場合は、サイトAのノードでこの値をyesに変更し、サービスを再起動します。

```
[root@jfs12 ~]# /grid/bin/crsctl set server css_critical yes
CRS-4416: Server attribute 'CSS_CRITICAL' successfully changed. Restart
Oracle High Availability Services for new value to take effect.

[root@jfs12 ~]# /grid/bin/crsctl stop crs
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'jfs12'
CRS-2673: Attempting to stop 'ora.crsd' on 'jfs12'
CRS-2790: Starting shutdown of Cluster Ready Services-managed resources on
server 'jfs12'
CRS-2673: Attempting to stop 'ora.ntap.ntappdb1.pdb' on 'jfs12'
…
CRS-2673: Attempting to stop 'ora.gipcd' on 'jfs12'
CRS-2677: Stop of 'ora.gipcd' on 'jfs12' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'jfs12' has completed
CRS-4133: Oracle High Availability Services has been stopped.

[root@jfs12 ~]# /grid/bin/crsctl start crs
CRS-4123: Oracle High Availability Services has been started.
```

障害シナリオ

概要

完全なSnapMirrorアクティブ同期アプリケーションアーキテクチャを計画するには、さまざまな計画的フェイルオーバーシナリオと計画外フェイルオーバーシナリオでSM-ASがどのように対応するかを理解する必要があります。

次の例では、サイトAが優先サイトとして設定されているとします。

レプリケーション接続の切断

SM-ASレプリケーションが中断されると、クラスタが反対側のサイトに変更をレプリケートできなくなるため、書き込みIOを完了できません。

サイト**A**（優先サイト）

優先サイトでのレプリケーションリンク障害の結果、レプリケーションリンクが本当に到達不能であると判断される前に、ONTAPがレプリケートされた書き込み処理を再試行するため、書き込みIO処理が約15秒間中断されます。15秒が経過すると、サイトAのシステムが読み取りと書き込みのIO処理を再開します。SANパスは変更されず、LUNはオンラインのままです。

サイト**B**

サイトBはSnapMirrorアクティブ同期優先サイトではないため、約15秒後にLUNパスが使用できなくなります。

ストレージシステムの障害

ストレージシステム障害の結果は、レプリケーションリンクが失われた場合とほぼ同じです。サバイバーサイトでは、IOが約15秒間停止します。その15秒が経過すると、IOは通常どおりそのサイトで再開されます。

メディエーターの停止

メディエーターサービスはストレージの処理を直接制御しません。クラスタ間の代替制御パスとして機能します。これは主に、スプリットブレインのリスクを伴わずにフェイルオーバーを自動化することを目的としています。通常運用時は、各クラスタがパートナーに変更内容をレプリケートするため、各クラスタはパートナークラスタがオンラインでデータを提供していることを確認できます。レプリケーションリンクに障害が発生すると、レプリケーションは停止します。

安全な自動フェイルオーバーを実現するためにメディエーターが必要になるのは、そうしないと、双方向通信の切断がネットワークの停止によるものか実際のストレージ障害によるものかをストレージクラスタが判断できないためです。

メディエーターは、パートナーの健常性を確認するための代替パスを各クラスタに提供します。シナリオは次のとおりです。

- クラスタがパートナーに直接接続できる場合は、レプリケーションサービスが動作しています。対処は不要です。
- 優先サイトがパートナーに直接またはメディエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離されてLUNパスがオフラインになっているとみなされます。その後、優先サイトでRPO=0の状態が解除され、読み取りI/Oと書き込みI/Oの両方の処理が続行されます。
- 非優先サイトがパートナーに直接接続できず、メディエーター経由で接続できる場合、そのサイトのパスはオフラインになり、レプリケーション接続が戻るまで待機します。
- 優先されないサイトがパートナーに直接、または動作中のメディエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離され、LUNパスがオフラインになったとみなされます。優先されないサイトは、RPO=0状態の解放に進み、読み取りI/Oと書き込みI/Oの両方の処理を続行します。レプリケーションソースの役割を引き継ぎ、新しい優先サイトになります。

メディエーターが完全に使用できない場合：

- 非優先サイトまたはストレージシステムの障害など、何らかの理由でレプリケーションサービスに障害が発生すると、優先サイトでRPO=0状態が解放され、読み取りおよび書き込みIO処理が再開されます。非優先サイトのパスがオフラインになります。
- 優先サイトに障害が発生すると、非優先サイトでは、反対側のサイトが本当にオフラインであることを確認できず、そのため非優先サイトがサービスを再開しても安全ではないため、システムが停止します。

サービスのリストア

サイト間の接続のリストアや障害が発生したシステムの電源投入などの障害が解決されると、SnapMirrorのアクティブな同期エンドポイントは、障害のあるレプリケーション関係の存在を自動的に検出してRPO=0状態に戻します。同期レプリケーションが再確立されると、障害が発生したパスは再びオンラインになります。

多くの場合、クラスタ化されたアプリケーションは障害が発生したパスの復帰を自動的に検出し、それらのアプリケーションもオンラインに戻ります。また、ホストレベルのSANスキャンが必要な場合や、アプリケーションを手動でオンラインに戻す必要がある場合もあります。それはアプリケーションとそれがどのように構成されているかによって異なり、一般的にそのようなタスクは簡単に自動化することができます。ONTAP自体は自己回復型であり、RPO=0のストレージ処理を再開するためにユーザの介入は不要です。

手動フェイルオーバー

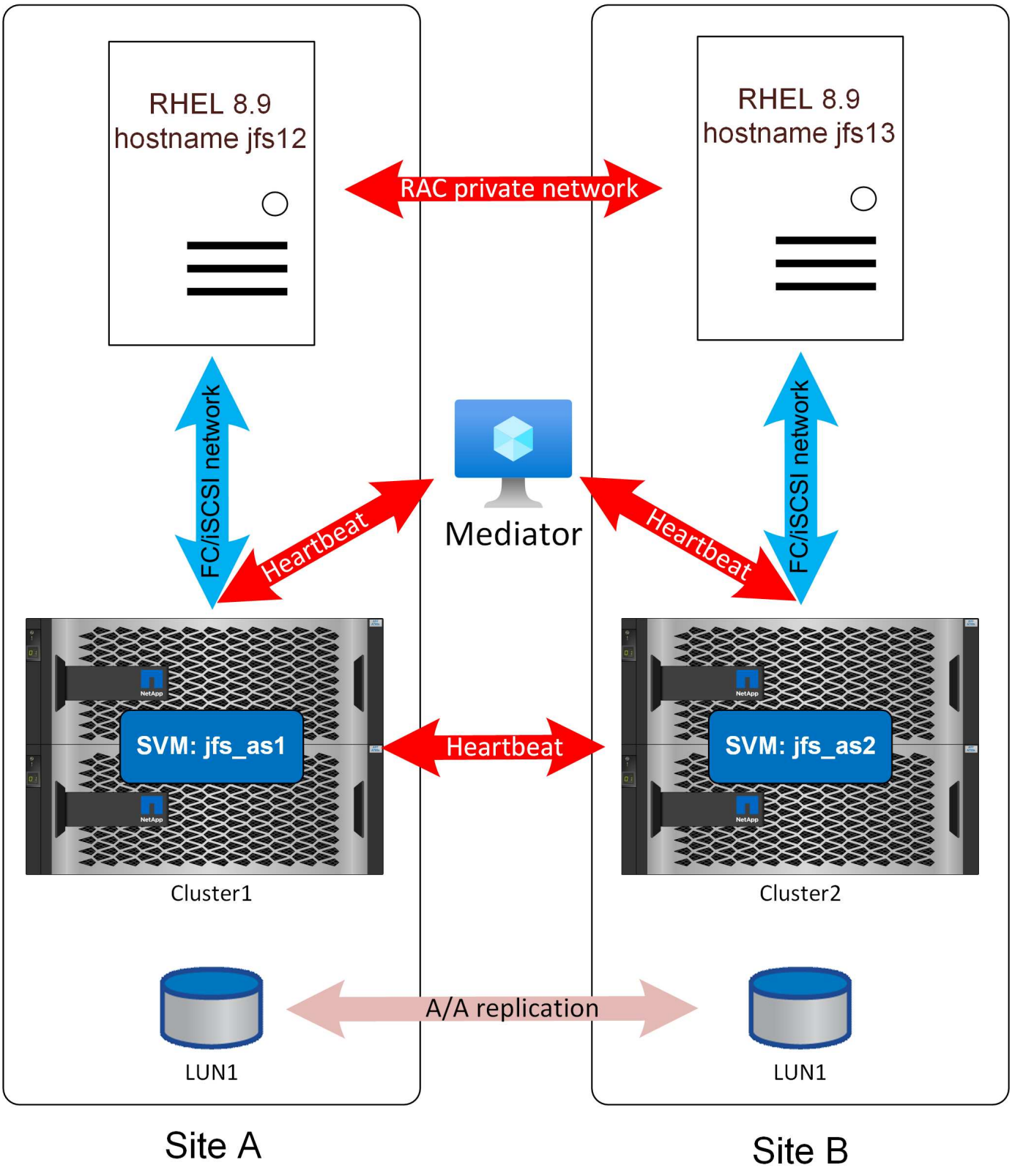
優先サイトを変更するには、簡単な操作が必要です。クラスタ間でレプリケーション動作の権限が切り替わるため、IOは1~2秒間停止しますが、それ以外の場合はIOには影響しません。

サンプルアーキテクチャ

このセクションで示す障害の詳細な例は、次のアーキテクチャに基づいています。

これは、SnapMirrorアクティブ同期でOracleデータベースを使用する場合のオプションの1つにすぎません。この設計は、いくつかのより複雑なシナリオを説明するために選択されました。

この設計では、サイトAがに設定されていると仮定し"優先サイト"ます。



RACインターコネクト障害

Oracle RACレプリケーションリンクが失われると、SnapMirror接続が切断されますが、デフォルトでタイムアウトが短くなる点が異なります。デフォルト設定では、Oracle RACノードはストレージ接続が失われてから200秒待機してから削除されますが、RAC

ネットワークハートビートが失われてからは30秒しか待機しません。

CRSメッセージは次のようになります。30秒のタイムアウトの経過が表示されます。CSS_CRITICALはサイトAにあるjfs12に設定されているため、これが存続するサイトとなり、サイトBのjfs13が削除されます。

```
2024-09-12 10:56:44.047 [ONMD(3528)]CRS-1611: Network communication with
node jfs13 (2) has been missing for 75% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 6.980 seconds
2024-09-12 10:56:48.048 [ONMD(3528)]CRS-1610: Network communication with
node jfs13 (2) has been missing for 90% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 2.980 seconds
2024-09-12 10:56:51.031 [ONMD(3528)]CRS-1607: Node jfs13 is being evicted
in cluster incarnation 621599354; details at (:CSSNM00007:) in
/gridbase/diag/crs/jfs12/crs/trace/onmd.trc.
2024-09-12 10:56:52.390 [CRSD(6668)]CRS-7503: The Oracle Grid
Infrastructure process 'crsd' observed communication issues between node
'jfs12' and node 'jfs13', interface list of local node 'jfs12' is
'192.168.30.1:33194;', interface list of remote node 'jfs13' is
'192.168.30.2:33621;'.
2024-09-12 10:56:55.683 [ONMD(3528)]CRS-1601: CSSD Reconfiguration
complete. Active nodes are jfs12 .
2024-09-12 10:56:55.722 [CRSD(6668)]CRS-5504: Node down event reported for
node 'jfs13'.
2024-09-12 10:56:57.222 [CRSD(6668)]CRS-2773: Server 'jfs13' has been
removed from pool 'Generic'.
2024-09-12 10:56:57.224 [CRSD(6668)]CRS-2773: Server 'jfs13' has been
removed from pool 'ora.NTAP'.
```

SnapMirror通信障害

SnapMirrorのアクティブな同期レプリケーションリンクの場合、クラスタが反対側のサイトに変更をレプリケートできないため、書き込みIOを完了できません。

サイト**A**

レプリケーションリンク障害が発生したサイトAでは、レプリケーションリンクが本当に動作不能であると判断される前に、ONTAPが書き込みをレプリケートしようとするため、書き込みIO処理が約15秒間中断されます。15秒が経過すると、サイトAのONTAPクラスタが読み取りと書き込みのIO処理を再開します。SANパスは変更されず、LUNはオンラインのままです。

サイト**B**

サイトBはSnapMirrorアクティブ同期優先サイトではないため、約15秒後にLUNパスが使用できなくなります。

レプリケーションリンクはタイムスタンプ15：19：44でカットされました。Oracle RACからの最初の警告は、200秒のタイムアウト（Oracle RACパラメータdisktimeoutで制御）が近づくと、100秒後に通知されます。

```
2024-09-10 15:21:24.702 [ONMD(2792)]CRS-1615: No I/O has completed after
50% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 99340 milliseconds.
2024-09-10 15:22:14.706 [ONMD(2792)]CRS-1614: No I/O has completed after
75% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 49330 milliseconds.
2024-09-10 15:22:44.708 [ONMD(2792)]CRS-1613: No I/O has completed after
90% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 19330 milliseconds.
2024-09-10 15:23:04.710 [ONMD(2792)]CRS-1604: CSSD voting file is offline:
/dev/mapper/grid2; details at (:CSSNM00058:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc.
2024-09-10 15:23:04.710 [ONMD(2792)]CRS-1606: The number of voting files
available, 0, is less than the minimum number of voting files required, 1,
resulting in CSSD termination to ensure data integrity; details at
(:CSSNM00018:) in /gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-10 15:23:04.716 [ONMD(2792)]CRS-1699: The CSS daemon is
terminating due to a fatal error from thread:
clssnmvDiskPingMonitorThread; Details at (:CSSSC00012:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-10 15:23:04.731 [OCSSD(2794)]CRS-1652: Starting clean up of CRSD
resources.
```

200秒の投票ディスクタイムアウトに達すると、このOracle RACノードはクラスタから削除され、リブートされます。

ネットワーク相互接続の全体的な障害

サイト間のレプリケーションリンクが完全に失われると、SnapMirrorアクティブ同期とOracle RAC接続の両方が中断されます。

Oracle RACのスプリットブレイン検出は、Oracle RACストレージのハートビートに依存します。サイト間の接続が失われてRACネットワークハートビートとストレージレプリケーションサービスの両方が同時に失われると、RACサイトはRACインターコネクトまたはRAC投票ディスクを介してサイト間通信できなくなります。その結果、ノード数が偶数になると、両方のサイトがデフォルト設定で削除される可能性があります。正確な動作は、イベントのシーケンス、RACネットワークおよびディスクハートビートポーリングのタイミングによって異なります。

2サイト停止のリスクには、2つの方法で対処できます。まず、"Tiebreaker"構成を使用できます。

3つ目のサイトが利用できない場合は、RACクラスタでmiscountパラメータを調整することでこのリスクに対処できます。デフォルトでは、RACネットワークハートビートタイムアウトは30秒です。通常、RACは障害が発生したRACノードを特定してクラスタから削除するために使用します。また、投票ディスクハートビートにも接続されています。

たとえば、Oracle RACとストレージレプリケーションサービスの両方でサイト間トラフィックを伝送するコンジットがバックホーでカットされると、30秒間のミスカウントのカウントダウンが開始されます。RAC優先サイトノードが30秒以内に反対サイトとの接続を再確立できない場合、および同じ30秒以内に反対サイト

が停止していることを投票ディスクを使用して確認できない場合、優先サイトノードも削除されます。その結果、データベースが完全に停止します。

ミスマウントポーリングが発生したタイミングによっては、30秒でSnapMirrorアクティブ同期がタイムアウトし、優先サイトのストレージでサービスが再開されるまでに30秒では不十分な場合があります。この30秒のウィンドウは増やすことができます。

```
[root@jfs12 ~]# /grid/bin/crsctl set css misscount 100
CRS-4684: Successful set of parameter misscount to 100 for Cluster
Synchronization Services.
```

この値を指定すると、優先サイト上のストレージシステムは、ミスカウントのタイムアウトが切れる前に処理を再開できます。その結果、LUNパスを削除したサイトのノードのみが削除されます。以下の例：

```
2024-09-12 09:50:59.352 [ONMD(681360)]CRS-1612: Network communication with
node jfs13 (2) has been missing for 50% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 49.570 seconds
2024-09-12 09:51:10.082 [CRSD(682669)]CRS-7503: The Oracle Grid
Infrastructure process 'crsd' observed communication issues between node
'jfs12' and node 'jfs13', interface list of local node 'jfs12' is
'192.168.30.1:46039;', interface list of remote node 'jfs13' is
'192.168.30.2:42037;'.
2024-09-12 09:51:24.356 [ONMD(681360)]CRS-1611: Network communication with
node jfs13 (2) has been missing for 75% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 24.560 seconds
2024-09-12 09:51:39.359 [ONMD(681360)]CRS-1610: Network communication with
node jfs13 (2) has been missing for 90% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 9.560 seconds
2024-09-12 09:51:47.527 [OHASD(680884)]CRS-8011: reboot advisory message
from host: jfs13, component: cssagent, with time stamp: L-2024-09-12-
09:51:47.451
2024-09-12 09:51:47.527 [OHASD(680884)]CRS-8013: reboot advisory message
text: oracssdagent is about to reboot this node due to unknown reason as
it did not receive local heartbeats for 10470 ms amount of time
2024-09-12 09:51:48.925 [ONMD(681360)]CRS-1632: Node jfs13 is being
removed from the cluster in cluster incarnation 621596607
```

Oracleサポートでは、設定の問題を解決するために、miscountパラメータやdisktimeoutパラメータを変更することを強く推奨していません。ただし、SANブート、仮想化、ストレージレプリケーションの構成など、多くの場合、これらのパラメータの変更は保証され、やむを得ない場合があります。たとえば、SANまたはIPネットワークの安定性に問題があり、その結果RACが削除された場合は、原因となっている問題を修正し、ミスカウントやdisktimeoutの値を加算しないでください。構成エラーに対処するためにタイムアウトを変更すると、問題がマスキングされ、問題が解決されません。基盤となるインフラの設計要素に基づいてRAC環境を適切に設定するためにこれらのパラメータを変更することは異なり、Oracleのサポートステートメントと一致しています。SANブートでは、disktimeoutに合わせて最大200までミスカウントを調整するのが一般的です。詳細については、を参照してください"リンクをクリックしてください"。

ストレージシステムまたはサイト障害の結果は、レプリケーションリンクが失われた場合とほぼ同じです。サバイバーサイトでは、書き込み時のIOポーズが約15秒になります。その15秒が経過すると、IOは通常どおりそのサイトで再開されます。

ストレージシステムのみが影響を受けた場合、障害が発生したサイトのOracle RACノードはストレージサービスを失い、削除とその後のリブートの前に同じ200秒のディスクタイムアウトカウントダウンを入力します。

```
2024-09-11 13:44:38.613 [ONMD(3629)]CRS-1615: No I/O has completed after
50% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 99750 milliseconds.
2024-09-11 13:44:51.202 [ORAAGENT(5437)]CRS-5011: Check of resource "NTAP"
failed: details at "(:CLSN00007:)" in
"/gridbase/diag/crs/jfs13/crs/trace/crsd_oraagent_oracle.trc"
2024-09-11 13:44:51.798 [ORAAGENT(75914)]CRS-8500: Oracle Clusterware
ORAAGENT process is starting with operating system process ID 75914
2024-09-11 13:45:28.626 [ONMD(3629)]CRS-1614: No I/O has completed after
75% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 49730 milliseconds.
2024-09-11 13:45:33.339 [ORAAGENT(76328)]CRS-8500: Oracle Clusterware
ORAAGENT process is starting with operating system process ID 76328
2024-09-11 13:45:58.629 [ONMD(3629)]CRS-1613: No I/O has completed after
90% of the maximum interval. If this persists, voting file
/dev/mapper/grid2 will be considered not functional in 19730 milliseconds.
2024-09-11 13:46:18.630 [ONMD(3629)]CRS-1604: CSSD voting file is offline:
/dev/mapper/grid2; details at (:CSSNM00058:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc.
2024-09-11 13:46:18.631 [ONMD(3629)]CRS-1606: The number of voting files
available, 0, is less than the minimum number of voting files required, 1,
resulting in CSSD termination to ensure data integrity; details at
(:CSSNM00018:) in /gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-11 13:46:18.638 [ONMD(3629)]CRS-1699: The CSS daemon is
terminating due to a fatal error from thread:
clssnmvDiskPingMonitorThread; Details at (:CSSSC00012:) in
/gridbase/diag/crs/jfs13/crs/trace/onmd.trc
2024-09-11 13:46:18.651 [OCSSD(3631)]CRS-1652: Starting clean up of CRSD
resources.
```

ストレージサービスが失われたRACノードのSANパスの状態は次のようになります。

```
oradata7 (3600a0980383041334a3f55676c697347) dm-20 NETAPP,LUN C-Mode
size=128G features='3 queue_if_no_path pg_init_retries 50' hwhandler='1
alua' wp=rw
|-+- policy='service-time 0' prio=0 status=enabled
| `- 34:0:0:18 sdam 66:96  failed faulty running
`-+- policy='service-time 0' prio=0 status=enabled
  `- 33:0:0:18 sdaj 66:48  failed faulty running
```

Linuxホストはパスの損失を200秒よりもはるかに早く検出しましたが、データベースに関しては、障害が発生したサイトのホストへのクライアント接続は、デフォルトのOracle RAC設定で200秒間フリーズします。フルデータベース処理は削除が完了するまで再開されません。

一方、反対側のサイトのOracle RACノードでは、もう一方のRACノードの損失が記録されます。それ以外の場合は、通常どおり動作し続けます。

```
2024-09-11 13:46:34.152 [ONMD(3547)]CRS-1612: Network communication with
node jfs13 (2) has been missing for 50% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 14.020 seconds
2024-09-11 13:46:41.154 [ONMD(3547)]CRS-1611: Network communication with
node jfs13 (2) has been missing for 75% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 7.010 seconds
2024-09-11 13:46:46.155 [ONMD(3547)]CRS-1610: Network communication with
node jfs13 (2) has been missing for 90% of the timeout interval.  If this
persists, removal of this node from cluster will occur in 2.010 seconds
2024-09-11 13:46:46.470 [OHASD(1705)]CRS-8011: reboot advisory message
from host: jfs13, component: cssmonit, with time stamp: L-2024-09-11-
13:46:46.404
2024-09-11 13:46:46.471 [OHASD(1705)]CRS-8013: reboot advisory message
text: At this point node has lost voting file majority access and
oracssdmonitor is rebooting the node due to unknown reason as it did not
receive local hearbeats for 28180 ms amount of time
2024-09-11 13:46:48.173 [ONMD(3547)]CRS-1632: Node jfs13 is being removed
from the cluster in cluster incarnation 621516934
```

メディエーター障害

メディエーターサービスはストレージの処理を直接制御しません。クラスタ間の代替制御パスとして機能します。これは主に、スプリットブレインのリスクを伴わずにフェイルオーバーを自動化することを目的としています。

通常運用時は、各クラスタがパートナーに変更内容をレプリケートするため、各クラスタはパートナークラスタがオンラインでデータを提供していることを確認できます。レプリケーションリンクに障害が発生すると、レプリケーションは停止します。

安全な自動運用を実現するためにメディエーターが必要になるのは、双方向通信の切断がネットワークの停止

によるものか実際のストレージ障害によるものかをストレージクラスタが判断できないためです。

メディエーターは、パートナーの健常性を確認するための代替パスを各クラスタに提供します。シナリオは次のとおりです。

- クラスタがパートナーに直接接続できる場合は、レプリケーションサービスが動作しています。対処は不要です。
- 優先サイトがパートナーに直接またはメディエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離されてLUNパスがオフラインになっているとみなされます。その後、優先サイトでRPO=0の状態が解除され、読み取りI/Oと書き込みI/Oの両方の処理が続行されます。
- 非優先サイトがパートナーに直接接続できず、メディエーター経由で接続できる場合、そのサイトのパスはオフラインになり、レプリケーション接続が戻るまで待機します。
- 優先されないサイトがパートナーに直接、または動作中のメディエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離され、LUNパスがオフラインになったとみなされます。優先されないサイトは、RPO=0状態の解放に進み、読み取りI/Oと書き込みI/Oの両方の処理を続行します。レプリケーションソースの役割を引き継ぎ、新しい優先サイトになります。

メディエーターが完全に使用できない場合：

- 何らかの理由でレプリケーションサービスに障害が発生すると、優先サイトでRPO=0状態が解放され、読み取りと書き込みのIO処理が再開されます。非優先サイトのパスがオフラインになります。
- 優先サイトに障害が発生すると、非優先サイトでは、反対側のサイトが本当にオフラインであることを確認できず、そのため非優先サイトがサービスを再開しても安全ではないため、システムが停止します。

サービスの復旧

SnapMirrorは自己回復型です。SnapMirrorのアクティブな同期では、レプリケーション関係に問題があることを自動的に検出し、RPO=0の状態に戻します。同期レプリケーションが再確立されると、パスは再びオンラインになります。

多くの場合、クラスタ化されたアプリケーションは障害が発生したパスの復帰を自動的に検出し、それらのアプリケーションもオンラインに戻ります。また、ホストレベルのSANスキャンが必要な場合や、アプリケーションを手動でオンラインに戻す必要がある場合もあります。

アプリケーションとその構成方法によって異なり、一般的にこのようなタスクは簡単に自動化できます。SnapMirrorのアクティブな同期自体は自己修復機能であり、電源と接続が復旧した時点でRPO=0のストレージ処理を再開するためにユーザの介入は必要ありません。

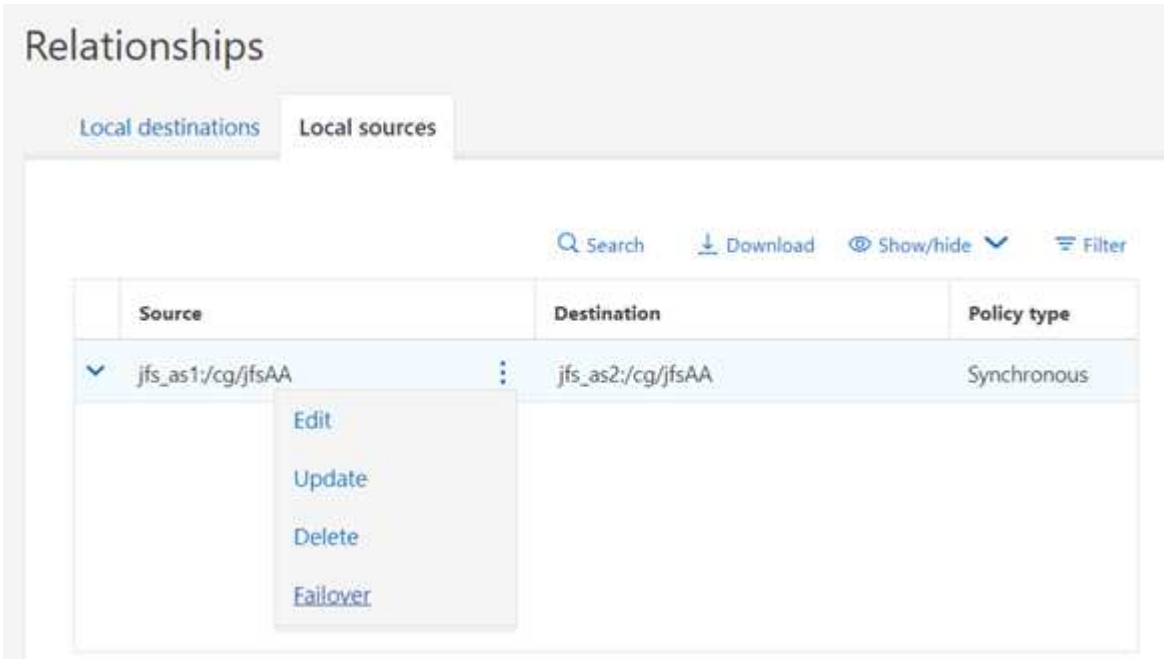
手動フェイルオーバー

「フェイルオーバー」という用語は、双方向のレプリケーションテクノロジであるため、SnapMirror Active Syncを使用したレプリケーションの方向を指していません。代わりに、「failover」とは、障害発生時にどのストレージシステムが優先サイトになるかを意味します。

たとえば、メンテナンスのためにサイトをシャットダウンする前やDRテストを実行する前に、フェイルオーバーを実行して優先サイトを変更できます。

優先サイトを変更するには、簡単な操作が必要です。クラスタ間でレプリケーション動作の権限が切り替わるため、IOは1~2秒間停止しますが、それ以外の場合はIOには影響しません。

GUI の例：



CLIを使用して元に戻す例：

```
Cluster2::> snapmirror failover start -destination-path jfs_as2:/cg/jfsAA
[Job 9575] Job is queued: SnapMirror failover for destination
"jfs_as2:/cg/jfsAA                        ".

Cluster2::> snapmirror failover show

Source    Destination                                              Error
Path      Path        Type     Status    start-time end-time     Reason
--------- ----------  -------- --------- ---------- ---------- ----------
jfs_as1:/cg/jfsAA
         jfs_as2:/cg/jfsAA
                     planned  completed 9/11/2024  9/11/2024
                                        09:29:22   09:29:32

The new destination path can be verified as follows:


Cluster1::> snapmirror show -destination-path jfs_as1:/cg/jfsAA

                           Source Path: jfs_as2:/cg/jfsAA
                        Destination Path: jfs_as1:/cg/jfsAA
                      Relationship Type: XDP
                Relationship Group Type: consistencygroup
                 SnapMirror Policy Type: automated-failover-duplex
                      SnapMirror Policy: AutomatedFailOverDuplex
                            Tries Limit: -
                           Mirror State: Snapmirrored
                    Relationship Status: InSync
```

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

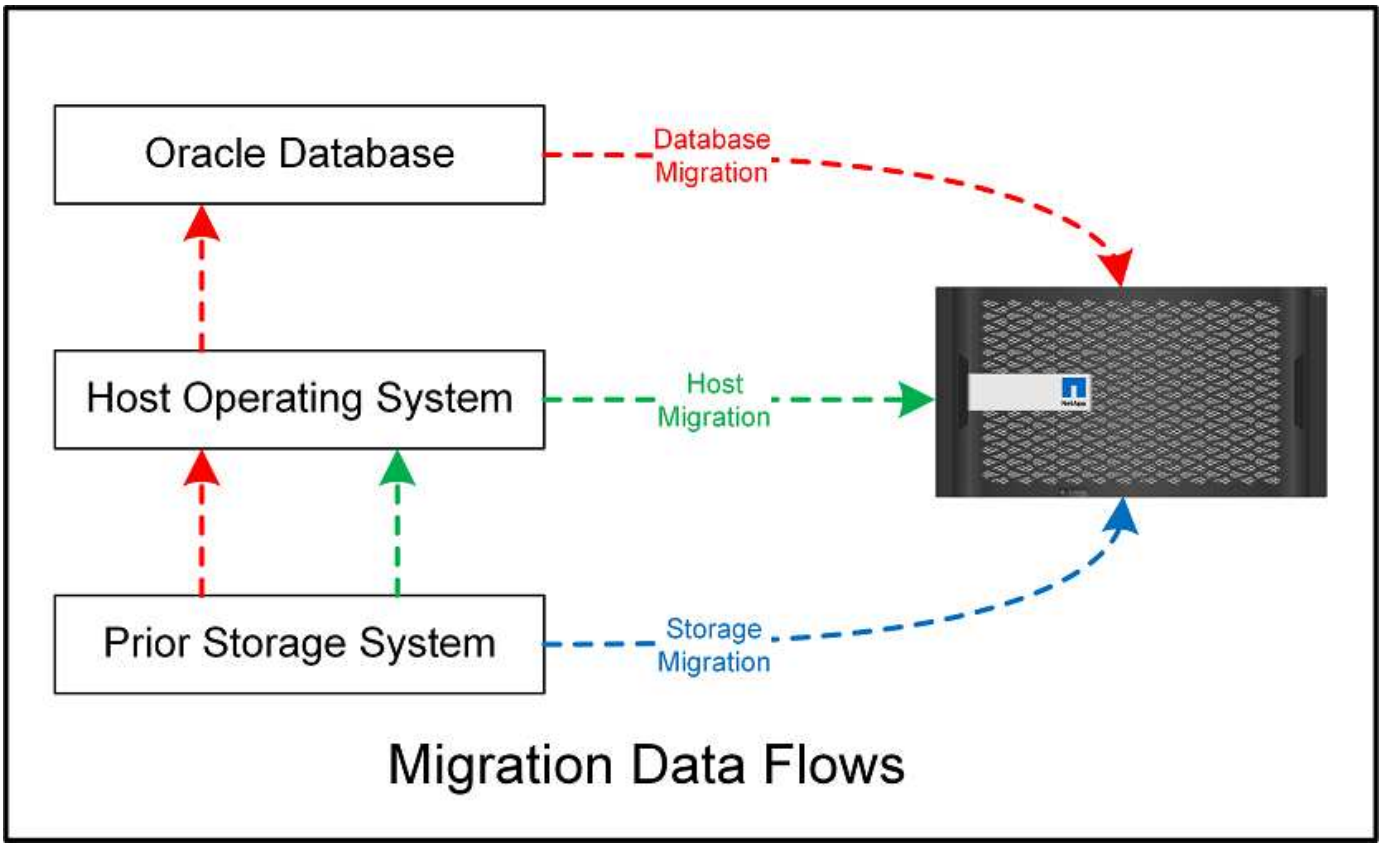
このドキュメントには、サンプルスクリプトが記載されています。これらのスクリプトは、ユーザによるミスの可能性を減らすために、移行のさまざまな側面を自動化するサンプル方法を提供します。スクリプトを使用すると、移行を担当するITスタッフの全体的な要求を軽減し、プロセス全体をスピードアップできます。これらのスクリプトはすべて、NetAppプロフェッショナルサービスとNetAppパートナーが実施した実際の移行プロジェクトを基に作成されています。このドキュメント全体で使用例が示されています。

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。



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て、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ヨウケン

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ア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アレイの機能は、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シリーズア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、新しいストレージア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時間で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。

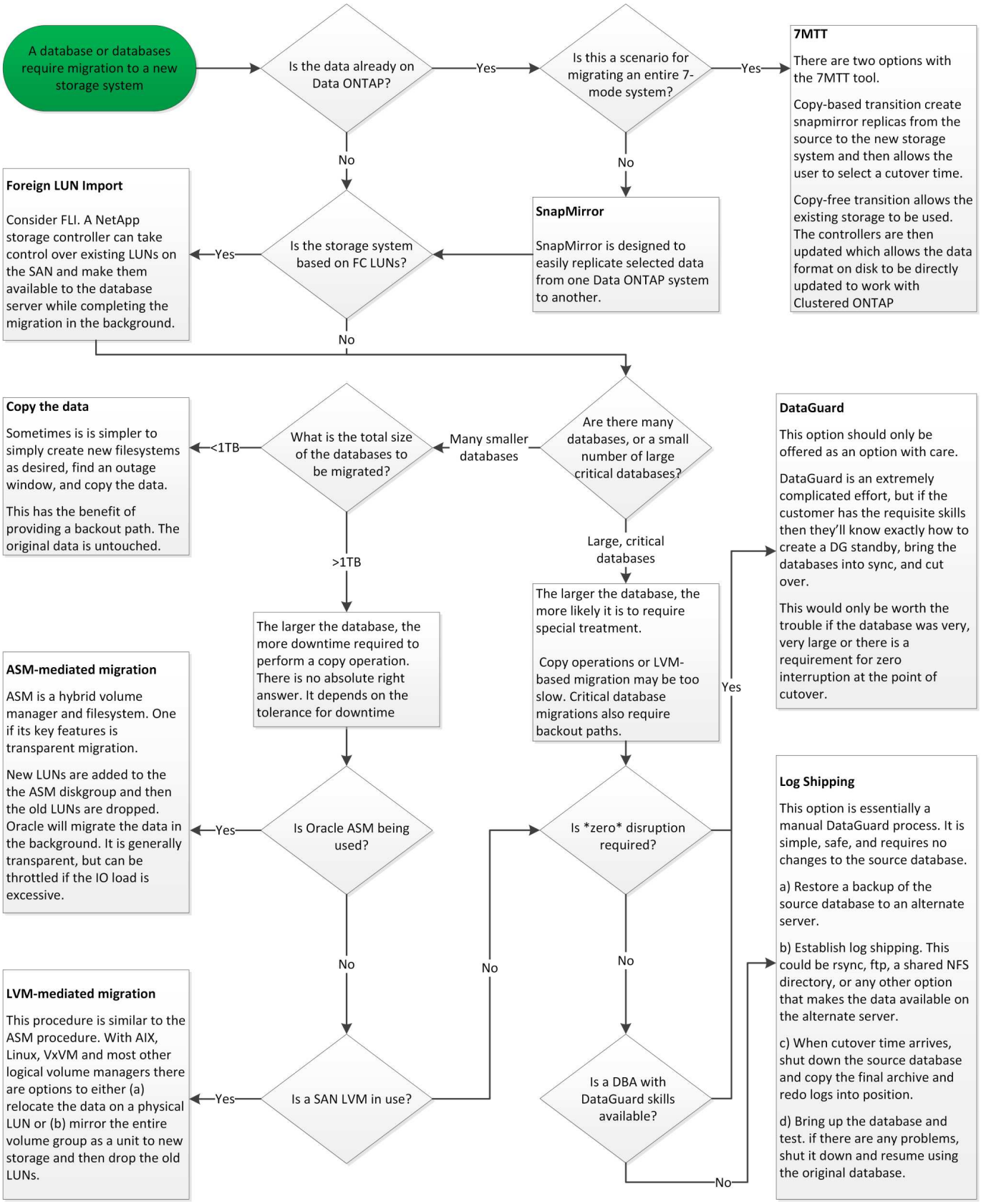
多くの場合、システム管理者とDBAには、物理ボリュームのデータの再配置、ミラーリングとミラーリングの解除、Oracle RMANを使用したデータのコピーなど、独自の方法があります。

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。

Oracle DataGuard

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、ストレージア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番目のサーバーを一時的なステージングサーバーとして使用できます。新しいストレージア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。バックアウトパスが必要な場合は、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て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 `pvscan` 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.ctrl';
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@jfsc3 tmp]$ scp oracle@jfsc1:/tmp/waffle.ctrl /tmp/
oracle@jfsc1's password:
waffle.ctrl                                               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@jfsc3 ~]$ . oraenv
ORACLE_SID = [oracle] ? WAFFLE
The Oracle base has been set to /orabin
[oracle@jfsc3 ~]$ cd $ORACLE_BASE
[oracle@jfsc3 orabin]$ cd admin
[oracle@jfsc3 admin]$ mkdir WAFFLE
[oracle@jfsc3 admin]$ cd WAFFLE
[oracle@jfsc3 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@jfsc3 admin]$ scp oracle@jfsc1:/tmp/waffle.tmp.pfile
$ORACLE_HOME/dbs/waffle.tmp.pfile
oracle@jfsc1'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@jfsc1 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/contr
ol02.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@jfsc3 arch]$ scp jfsc1:/logs/WAFFLE/arch/* ./
oracle@jfsc1'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に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@jfsc1 arch]# cat /etc/rsyncd.conf
[waffle.arch]
   uid=oracle
   gid=dba
   path=/logs/WAFFLE/arch
   read only = true
[root@jfsc1 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で定期的に実行する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@jfsc3 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
```

カットオーバー

新しい環境にカットオーバーする準備ができたら、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@jfsc3 logs]$ scp jfsc1:/logs/WAFFLE/redo/* /redo/
   oracle@jfsc1'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/1gr6c161_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/1hr6c164_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@jfsc2 ~]$ mkdir /oradata/pancake
[oracle@jfsc2 ~]$ mkdir /logs/pancake
[oracle@jfsc2 ~]$ cd /orabin/admin
[oracle@jfsc2 admin]$ mkdir PANCAKE
[oracle@jfsc2 admin]$ cd PANCAKE
[oracle@jfsc2 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pfile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
```

初期ログレプリケーション

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@jfsc1 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@jfsc1 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@jfsc2 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
```

カットオーバー

新しい環境にカットオーバーする準備ができたら、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@jfsc2 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が、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を削除しようとしましたが、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いため、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,rsize=65536,wsize=65536,addr=172.20.101.10)
ontap-nfs1:/host1_logs on /logs type nfs
(ro,bg,vers=3,rsize=65536,wsize=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、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,rsize=65536,wsize=65536,addr=172.20.101.10)
ontap-nfs1:/host1_oradata_new on /oradata type nfs
(rw,bg,vers=3,rsize=65536,wsize=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の状態をリバートして、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。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テイ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
```

このサーバーには+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カラASMへノコヒイ

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を削除しようとしましたが、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
ASMCMD> umount DATA
ASMCMD>
```

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/control
02.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@jfsc1 ~]$ cd $ORACLE_HOME/dbs
[oracle@jfsc1 dbs]$ cat initTOAST.ora
SPFILE='+ASM0/TOAST/spfileTOAST.ora'
[oracle@jfsc1 dbs]$ mv initTOAST.ora initTOAST.ora.prev
[oracle@jfsc1 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.ctl
   output file name=/logs/TOAST/redo/control02.ctl
   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を削除しようとしましたが、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@jfsc1 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
ASMCMD> umount DATA
ASMCMD>
```

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ることを示します。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@jfsc1 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のエクステントを新しい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、以前のストレージア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$HuVWBi
FOR-1.14 800DT$HuVWBh
FOR-1.15 800DT$HuVWBj
FOR-1.16 800DT$HuVWBk
FOR-1.17 800DT$HuVWBm
FOR-1.18 800DT$HuVWBl
FOR-1.19 800DT$HuVWBo
FOR-1.20 800DT$HuVWBn
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$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -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で使用できるようになり、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なりました。

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$HuVWBX /vol/new_asm/LUN1 6
vserver1  800DT$HuVWBY /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$HuVWBh /vol/new_lvm/LUN2 4
vserver1  800DT$HuVWBi /vol/new_lvm/LUN3 3
vserver1  800DT$HuVWBj /vol/new_lvm/LUN4 3
vserver1  800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1  800DT$HuVWBl /vol/new_lvm/LUN6 4
vserver1  800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1  800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1  800DT$HuVWBo /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  sdbr 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 fpCzdLTuKfy2xDZjai1NliJh3TjLUBiT: 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テハイスノサイスキヤン

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。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。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$HuVWBX /vol/new_asm/LUN1 completed
vserver1 800DT$HuVWBY /vol/new_asm/LUN2 completed
vserver1 800DT$HuVWBZ /vol/new_asm/LUN3 completed
vserver1 800DT$HuVWBa /vol/new_asm/LUN5 completed
vserver1 800DT$HuVWBb /vol/new_asm/LUN6 completed
vserver1 800DT$HuVWBc /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$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWBi /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWBl /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWBo /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";}
```

```
 elsif ($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,rsize=65536,wsize=65536
0 0
cluster01:/vol_logs /logs nfs rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_newlogs /newlogs nfs rw,bg,vers=3,rsize=65536,wsize=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@jfsc3 scripts]# ./swap.fs.pl /oradata,/neworadata
/neworadata unmounted
/oradata unmounted
Mounted updated /oradata
[root@jfsc3 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,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_neworadata /oradata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_logs /logs nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_newlogs /newlogs nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_newlogs /logs nfs rw,bg,vers=3,rsize=65536,wsize=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 - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$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
$command
EOF2
        `;};
return @lines}
print "\n";
@out=dosql('select name from v\\\\\$datafile;');
print "$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/' ||
regexp_replace(lfrec.member,'^.*/','') || '''');
        else
            dbms_output.put(',''' || lfrec.member || ''', ');
            dbms_output.put('''/NEW_PATH/' ||
regexp_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;
```

その他の注意事項

パフォーマンスの最適化とベンチマーク

データベースストレージのパフォーマンスを正確にテストすることは、非常に複雑な課題です。次の問題について理解しておく必要があります。

- IOPSとスループット
- フォアグラウンドI/O処理とバックグラウンドI/O処理の違い
- データベースへのレイテンシの影響
- ストレージのパフォーマンスにも影響する多数のOSとネットワーク設定

また、ストレージデータベース以外のタスクについても考慮する必要があります。ストレージパフォーマンスがパフォーマンスの制限要因ではなくなったため、ストレージパフォーマンスを最適化しても有益なメリットは得られなくなります。

現在、データベースユーザの大半がオールフラッシュアレイを選択していることから、新たな考慮事項がいくつか生まれています。たとえば、2ノードのAFF A900システムでパフォーマンスをテストする場合を考えてみましょう。

- 読み取り/書き込み比率が80対20のA900ノードでは、レイテンシが150 µ sマークを超える前に、100万を超えるランダムデータベースIOPSを達成できます。これは、ほとんどのデータベースで現在必要とされているパフォーマンスをはるかに超えているため、予想される改善を予測することは困難です。ストレージがボトルネックになることはほとんどありません。
- ネットワーク帯域幅は、パフォーマンス上の制約の原因としてますます一般的になっています。たとえば、回転式ディスクソリューションはI/Oレイテンシが非常に高いため、データベースパフォーマンスのボトルネックになることがよくあります。オールフラッシュアレイでレイテンシの制限が取り除かれると、多くの場合、その障壁はネットワークに移ります。これは、真のネットワーク接続を可視化することが困難な仮想環境やブレードシステムで特に顕著です。帯域幅の制限のためにストレージシステム自体をフルに活用できない場合、パフォーマンステストが複雑になる可能性があります。
- オールフラッシュアレイのレイテンシが劇的に改善されるため、オールフラッシュアレイと回転式ディスクを搭載したアレイのパフォーマンスを比較することは一般的に不可能です。通常、テスト結果は意味がありません。
- オールフラッシュアレイでピーク時のIOPSパフォーマンスを比較することは、データベースがストレージI/Oの制約を受けないため、あまり有益なテストではありません。たとえば、あるアレイが50万IOPSを維持でき、別のアレイが30万IOPSを維持できるとします。データベースの処理時間の99%がCPU処理に費やされている場合、この違いは現実の世界では無関係です。ワークロードがストレージアレイのすべての機能を利用することはありません。一方、ストレージアレイの能力を最大限に引き出すことが期待される統合プラットフォームでは、ピーク時のIOPS性能が重要になる場合があります。
- どのストレージテストでも、レイテンシとIOPSを常に考慮してください。市場に出回っているストレージアレイの多くは、非常に高いIOPSを謳っていますが、このレベルのIOPSはレイテンシによって役に立たなくなります。オールフラッシュアレイの一般的なターゲットは1ミリ秒です。テストの優れた方法は、可能な最大IOPSを測定することではなく、平均レイテンシが1ミリ秒を超える前にストレージアレイが維持できるIOPSを特定することです。

Oracle自動ワークロードリポジトリとベンチマーク

Oracleのパフォーマンス比較のゴールドスタンダードは、Oracle Automatic Workload Repository（AWR）レポートです。

AWRレポートには複数のタイプがあります。ストレージの観点から見ると、 `awrrpt.sql` コマンドは、特定のデータベースインスタンスを対象としており、レイテンシに基づいてストレージI/Oイベントを内訳表示する詳細なヒストグラムが含まれているため、最も包括的で価値があります。

2つのパフォーマンスアレイを比較するには、各アレイで同じワークロードを実行し、ワークロードを正確に対象とするAWRレポートを作成するのが理想的です。非常に長時間のワークロードの場合は、開始時間と停止時間を含む経過時間を含む単一のAWRレポートを使用できますが、AWRデータを複数のレポートとして分割することを推奨します。たとえば、バッチジョブが午前0時から午前6時まで実行された場合は、午前0時から午前1時、午前1時から午前2時などの1時間のAWRレポートを作成します。

それ以外の場合は、非常に短いクエリを最適化する必要があります。最適なオプションは、クエリの開始時に作成されたAWRスナップショットと、クエリの終了時に作成された2番目のAWRスナップショットに基づくAWRレポートです。データベースサーバは、分析中のクエリのアクティビティを隠すバックグラウンドアクティビティを最小限に抑えるために、それ以外の場合は静かにしておく必要があります。

AWRレポートを使用できない場合は、代わりにOracle Statspackレポートを使用することを推奨します。AWRレポートとほとんど同じI/O統計情報が含まれています。

Oracle AWRとトラブルシューティング

AWRレポートは、パフォーマンスの問題を分析するための最も重要なツールでもあります。

ベンチマークと同様に、パフォーマンスのトラブルシューティングでは、特定のワークロードを正確に測定する必要があります。可能な場合は、パフォーマンスの問題をNetAppサポートセンターに報告するとき、または新しい解決策についてNetAppまたはパートナーアカウントチームと協力するときにAWRデータを提供してください。

AWRデータを提供する場合は、次の要件を考慮してください。

- を実行します `awrrpt.sql` レポートを生成するコマンド。出力はテキストまたはHTMLのいずれかになります。
- Oracle Real Application Clusters（RAC）を使用する場合は、クラスタ内の各インスタンスについてAWRレポートを生成します。
- 問題が発生した特定の時間をターゲットにします。AWRレポートの最大許容経過時間は、通常1時間です。問題が複数時間続く場合、またはバッチジョブなどの複数時間の操作を伴う場合は、分析対象の期間全体をカバーする複数の1時間のAWRレポートを提供します。
- 可能であれば、AWRスナップショット間隔を15分に調整します。この設定では、より詳細な分析を実行できます。これには、次の追加の実行も必要です。 `awrrpt.sql` 15分間隔ごとにレポートを作成します。
- 実行中のクエリが非常に短い場合は、操作の開始時に作成されたAWRスナップショットと、操作の終了時に作成された2つ目のAWRスナップショットに基づいてAWRレポートを提供します。それ以外の場合は、分析中の操作のアクティビティを隠すバックグラウンドアクティビティを最小限に抑えるために、データベースサーバは静かにしておく必要があります。
- パフォーマンスの問題が特定の時間に報告され、他の時間には報告されない場合は、比較のために優れたパフォーマンスを示す追加のAWRデータを提供します。

キャリブレーション_IO

。 `calibrate_io` コマンドは、ストレージシステムのテスト、比較、ベンチマークには使用しないでください。Oracleのドキュメントに記載されているように、この手順はストレージのI/O機能を調整します。

キャリブレーションはベンチマークと同じではありません。このコマンドの目的は、問題I/Oを使用して、データベース処理を調整し、ホストに対して実行されるI/Oのレベルを最適化することで効率を向上させることです。これは、によって実行されるI/Oのタイプが `calibrate_io` 処理が実際のデータベースユーザI/Oを表しているわけではありません。結果は予測不可能であり、再現さえできないこともよくあります。

SLOB2

Silly Little Oracle BenchmarkであるSLOB2は、データベースのパフォーマンス評価に好まれるツールになりました。Kevin Clossonによって開発され、次のサイトで入手できます。 "https://kevinclosson.net/slob/"。インストールと設定には数分かかり、実際のOracleデータベースを使用してユーザ定義の表領域にI/Oパターンを生成します。オールフラッシュアレイをI/Oで飽和状態にすることができる数少ないテストオプションの1つです。生成されるI/Oのレベルをはるかに低くして、IOPSは低くてもレイテンシの影響を受けやすいストレージワークロードをシミュレートする場合にも役立ちます。

スイングベンチ

Swingbenchはデータベースのパフォーマンスをテストするのに役立ちますが、ストレージに負荷がかかるような方法でSwingbenchを使用することは非常に困難です。NetAppでは、Swingbenchによるテストで、AFF

アレイに多大な負荷をかけるのに十分なI/Oが生成されたことはありません。一部のケースでは、Order Entry Test（OET）を使用してレイテンシの観点からストレージを評価できます。これは、データベースに特定のクエリに対する既知のレイテンシの依存関係がある場合に役立ちます。オールフラッシュアレイの潜在的なレイテンシを実現できるように、ホストとネットワークを適切に設定する必要があります。

HammerDB

HammerDBは、TPC-CやTPC-Hのベンチマークなどをシミュレートするデータベーステストツールです。テストを適切に実行するために十分な大きさのデータセットを構築するには、多くの時間がかかることがありますが、OLTPアプリケーションやデータウェアハウスアプリケーションのパフォーマンスを評価するための効果的なツールになる可能性があります。

オリオン

Oracle OrionツールはOracle 9で一般的に使用されていましたが、さまざまなホストオペレーティングシステムの変更に対応するためにメンテナンスが行われていません。OSやストレージ構成との互換性がないため、Oracle 10やOracle 11で使用されることはほとんどありません。

Oracleはこのツールを書き直し、Oracle 12cにデフォルトでインストールされます。この製品は改良され、実際のOracleデータベースと同じ呼び出しの多くを使用しますが、コードパスやI/O動作はOracleで使用されているものとまったく同じではありません。たとえば、ほとんどのOracle I/Oは同期的に実行されます。つまり、I/O処理はフォアグラウンドで完了するため、I/Oが完了するまでデータベースは停止します。ストレージシステムをランダムI/Oでフラッディングするだけでは、実際のOracle I/Oが再現されるわけではなく、ストレージアレイを比較したり、構成変更の影響を測定したりする直接的な方法もありません。

とはいえ、特定のホスト/ネットワーク/ストレージ構成の最大パフォーマンスの一般的な測定や、ストレージシステムの健全性の測定など、Orionのユースケースもあります。綿密なテストを実施すれば、Orionの使用可能なテストを考案して、ストレージアレイを比較したり、構成変更の影響を評価したりすることができます。ただし、パラメータにIOPS、スループット、レイテンシを考慮し、現実的なワークロードを忠実にレプリケートしようとする必要があります。

古い**NFSv3**ロック

Oracleデータベースサーバがクラッシュすると、再起動時に古いNFSロックで問題が発生する可能性があります。この問題は、サーバの名前解決を注意深く設定することで回避できます。

この問題は、ロックの作成とロックの解除で使用される名前解決方法がわずかに異なるために発生します。Network Lock Manager（NLM；ネットワークロックマネージャ）とNFSクライアントの2つのプロセスが関係しています。NLMでは、 `uname n` ホスト名を確認するには、 `rpc.statd` プロセスの用途 `gethostbyname()`。OSが古いロックを適切に解除するには、これらのホスト名が一致している必要があります。たとえば、ホストで所有されているロックが検索されているとします。 `dbserver5`が、ロックはホストによって次のように登録されています。 `dbserver5.mydomain.org`。状況 `gethostbyname()` と同じ値を返さない `uname –a`をクリックすると、ロック解除プロセスが成功しませんでした。

次のサンプルスクリプトは、名前解決が完全に一貫しているかどうかを検証します。

```
#! /usr/bin/perl
$uname=`uname -n`;
chomp($uname);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $uname;
print "uname -n yields: $uname\n";
print "gethostbyname yields: $name\n";
```

状況 `gethostbyname` 一致しません `uname`古いロックが使用されている可能性があります。たとえば、次の結果は潜在的な問題を示しています。

```
uname -n yields: dbserver5
gethostbyname yields: dbserver5.mydomain.org
```

解決策は通常、に表示されるホストの順序を変更することによって検出されます。 `/etc/hosts`。たとえば、hostsファイルに次のエントリが含まれているとします。

```
10.156.110.201  dbserver5.mydomain.org dbserver5 loghost
```

この問題を解決するには、完全修飾ドメイン名と短いホスト名の表示順序を変更します。

```
10.156.110.201  dbserver5 dbserver5.mydomain.org loghost
```

`gethostbyname()` では、shortを返します。 `dbserver5` ホスト名（の出力に一致） `uname`。したがって、ロックはサーバクラッシュ後に自動的にクリアされます。

WAFLアライメントの検証

優れたパフォーマンスを実現するには、WAFLを正しくアライメントすることが重要です。ONTAPはブロックを4KB単位で管理しますが、すべての処理がONTAPで4KB単位で実行されるわけではありません。実際、ONTAPはさまざまなサイズのブロック処理に対応しますが、基盤となる計算処理はWAFLによって4KB単位で管理されます。

「アライメント」という用語は、Oracle I/Oがこれらの4KBユニットにどのように対応するかを意味します。パフォーマンスを最適化するには、ドライブ上の2つの4KB WAFL物理ブロックにOracleの8KBブロックが配置されている必要があります。1つのブロックが2KBずれて配置されると、このブロックは1つの4KBブロックの半分、別の4KBブロック全体、3つ目の4KBブロックの半分に配置されます。このように配置すると、パフォーマンスが低下します。

NASファイルシステムでは、アライメントは問題になりません。Oracleデータファイルは、Oracleブロックのサイズに基づいてファイルの先頭にアライメントされます。したがって、8KB、16KB、32KBのブロックサイズは常にアライメントされます。すべてのブロック処理は、ファイルの先頭から4KB単位でオフセットされます。

一方、LUNの開始位置には何らかのドライバヘッダーやファイルシステムのメタデータが含まれているため、

オフセットが作成されます。最新のOSでは、アライメントが問題になることはほとんどありません。最新のOSは、標準の4KBセクターを使用する物理ドライブ向けに設計されており、パフォーマンスを最適化するためにI/Oを4KBの境界にアライメントする必要があるためです。

ただし、いくつかの例外があります。4KB I/O用に最適化されていない古いOSからデータベースが移行された場合や、パーティション作成時のユーザエラーによって4KB単位以外のオフセットが発生した場合があります。

以下はLinux固有の例ですが、手順はどのOSにも適用できます。

アライメント済み

次の例は、パーティションが1つの単一のLUNでアライメントチェックを示しています。

まず、ドライブで使用可能なすべてのパーティションを使用するパーティションを作成します。

```
[root@host0 iscsi]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xb97f94c1.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240):
Using default value 10240
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
[root@host0 iscsi]#
```

アライメントは、次のコマンドを使用して数学的にチェックできます。

```
[root@host0 iscsi]# fdisk -u -l /dev/sdb
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              32    20971519    10485744   83  Linux
```

出力は、単位が512バイトで、パーティションの開始が32ユニットであることを示しています。これは32 x 512 = 16、834バイトで、これは4KBのWAFLブロックの倍数です。このパーティションは正しくアライメントされています。

アライメントが正しいことを確認するには、次の手順を実行します。

1. LUNのUniversally Unique Identifier（UUID）を特定します。

```
FAS8040SAP::> lun show -v /vol/jfs_luns/lun0
              Vserver Name: jfs
                  LUN UUID: ed95d953-1560-4f74-9006-85b352f58fcd
                    Mapped: mapped`                 `
```

2. ONTAPコントローラでノードシェルを開始します。

```
FAS8040SAP::> node run -node FAS8040SAP-02
Type 'exit' or 'Ctrl-D' to return to the CLI
FAS8040SAP-02> set advanced
set not found.  Type '?' for a list of commands
FAS8040SAP-02> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by NetApp
         personnel.
```

3. 最初の手順で特定したターゲットUUIDで統計収集を開始します。

```
FAS8040SAP-02*> stats start lun:ed95d953-1560-4f74-9006-85b352f58fcd
Stats identifier name is 'Ind0xffffff08b9536188'
FAS8040SAP-02*>
```

4. I/Oを実行します。次のツールを使用することが重要です。 `iflag` I/Oが同期でバッファされていないことを確認する引数。

このコマンドには十分注意してください。の反転 `if` および `of` 引数はデータを破棄します。

```
[root@host0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000
bs=4096
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. 統計を停止し、アライメントのヒストグラムを表示します。すべてのI/Oが `.0` Bucket。4KBのブロック境界にアライメントされたI/Oを示します。

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff08b9536188
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:186%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
```

ミスアライメント状態です

次の例は、ミスアライメントI/Oを示しています。

1. 4KBの境界にアライメントされないパーティションを作成します。最新のOSでは、これはデフォルトの動作ではありません。

```
[root@host0 iscsi]# fdisk -u /dev/sdb
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (32-20971519, default 32): 33
Last sector, +sectors or +size{K,M,G} (33-20971519, default 20971519):
Using default value 20971519
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

2. パーティションは、デフォルトの32ではなく33セクターオフセットで作成されています。で説明されている手順を繰り返します。 "アライメント済み"。ヒストグラムは次のように表示されます。

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:136%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_partial_blocks:31%
```

ミスアライメントは明らかです。I/Oの大部分は* *.1 バケット。想定されるオフセットに一致します。パーティションが作成されたときに、最適化されたデフォルトよりも512バイト先のデバイスに移動されました。これは、ヒストグラムが512バイトオフセットされることを意味します。

また、も参照してください `read_partial_blocks` 統計がゼロ以外の場合は、実行されたI/Oが4KBブロック全体を一杯にしなかったことを意味します。

Redoロギング

ここで説明する手順はデータファイルに適用できます。OracleのREDOログとアーカイブログでは、I/Oパターンが異なります。たとえば、Redoロギングでは、単一ファイルを繰り返し上書きします。デフォルトの512バイトのブロックサイズを使用する場合、書き込み統計は次のようになります。

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-
9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.0:12%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.1:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.3:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.4:13%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.5:6%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.6:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.7:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_partial_blocks:85%
```

I/Oはすべてのヒストグラムバケットに分散されますが、これはパフォーマンス上の問題ではありません。ただし、4KBのブロックサイズを使用すると、Redoロギング率が非常に高くなる場合があります。この場合は、RedoロギングLUNが適切にアライメントされていることを確認することを推奨します。ただし、これは優れたパフォーマンスにとってデータファイルのアライメントほど重要ではありません。