テクニカル レポート

ONTAP上のOracleデータベース

NetApp
Jeffrey Steiner
2021年5月| TR-3633

概要

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

目次

表一覧

図一覧

はじめに

NetApp® ONTAP®は、インライン圧縮、ハードウェアの無停止アップグレード、外部ストレージアレイからのLUNインポートなどの機能を標準搭載した強力なデータ管理ソフトウェアです。最大24ノードのクラスタ構成が可能で、同時にNFS、SMB、iSCSI、FC、NVMEM Express（NVMe）の各プロトコルを通じてデータを提供できます。さらに、NetApp Snapshot™ テクノロジは、数万のオンラインバックアップと完全に運用可能なデータベースクローンを作成するための基盤となります。

ONTAPソフトウェアの豊富な機能セットに加えて、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、ユーザにはさまざまな要件があります。NetAppストレージは、VMware ESXの仮想環境で稼働する約6,000のデータベースから、996TBのシングル インスタンス データ ウェアハウス（規模は拡大中）まで、あらゆる環境に導入されています。そのため、NetAppストレージを基盤にしてOracleデータベースを構築するにあたって、明確なベストプラクティスというものはほとんどありません。

本ドキュメントでは、NetAppストレージ環境でOracleデータベースを運用するにあたっての要件を、2つの方法で解説します。1つは、明確なベストプラクティスがある場合、それを具体的に紹介する方法。もう1つは、設計の際に考慮すべき多数の事柄を確認していく方法です。Oracle向けストレージ ソリューションの設計者は、それぞれのビジネス要件を基に、この考慮事項に対処しなければなりません。

本ドキュメントではまず、すべての環境に共通する一般的な考慮事項を説明し、続いて、使用する仮想化ソリューションやOSごとに固有の推奨事項を解説します。ファイルシステムのレイアウトの選択やNFSロックの解除など、特殊なトピックについては付録で取り上げます。

詳細については、次のリソースを参照してください。

- TR-4591：『Database Data Protection』
- TR-4592：『Oracle on MetroCluster』
- TR-4534：『Migration of Oracle Databases to NetApp Storage Systems』

ONTAPソフトウェア

ONTAPは高度なデータ保護と管理の基盤をなすものですが、ソフトウェアのみで構成されます。選択できるタイプはいくつかあります。

- AFFおよびFAS上のONTAP
- NetApp ONTAP Select
- NetApp Cloud Volumes ONTAP

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

AFFおよびFASコントローラを搭載したONTAP

ONTAP搭載のAFF / FAS物理コントローラは、パフォーマンスとデータの制御性という点では最も優れたソリューションです。これは、20年以上にわたって何千ものお客様が利用してきた標準的なオプションです。ONTAPは、ミッションクリティカルな3つのデータベースから6万のデータベースを使用するサービスプロバイダ環境、ペタバイト規模のデータベースの瞬時のリストア、単一のデータベースの数百のクローンを使用するデータベースサービス（DBaaS）まで、あらゆる環境に対応するソリューションを提供します。

ONTAP Select

ONTAP Selectは、お客様の仮想インフラ上で動作し、ノーブランドのハードウェアに内蔵されたドライブにONTAPインテリジェンスとデータファブリック接続を提供します。また、ONTAP Selectを使用すると、ONTAPソフトウェアとゲストオペレーティングシステムで物理ハードウェアを共有して、高度なコンバージドインフラを構築できます。ONTAPを基盤にしたOracleのベストプラクティスには何も影響しません。一番の懸念はパフォーマンスですが、ONTAP Selectはこの点でも十分な機能を提供します。

ハイエンドのAFFシステムの最大パフォーマンスには及びませんが、データベースに30万IOPSが求められることはまずありません。一般的なデータベースなら、ONTAP Selectで達成できる約5,000～10,000IOPSで十分です。ほとんどのデータベースはストレージのIOPSよりもレイテンシから受ける影響の方が大きく、この問題はONTAP SelectをSSDに導入することで対応可能です。

Cloud Volumes ONTAP

Cloud Volumes ONTAPはONTAP Selectと類似していますが、ハイパー スケーラ クラウド環境で動作し、ハイパー スケーラ ストレージ ボリュームにインテリジェンスとデータ ファブリック接続を提供する点が異なります。ONTAPを基盤にしたOracleのベストプラクティスには何も影響しません。主な考慮事項は、パフォーマンスと、より低いエクステント コストです。

Cloud Volumes ONTAPのパフォーマンスは、クラウド プロバイダが管理する基盤のボリュームのパフォーマンスから部分的な制約を受けますが、その結果、ストレージの管理性が向上し、のキャッシュ機能によってパフォーマンスが向上する場合もあります。ただしIOPSとレイテンシに関してはパブリック クラウド プロバイダ次第なので、多少の制約が常に発生します。これらの制限は、データベースのパフォーマンスが許容できないことを意味するものではありません。単純に、最大パフォーマンスが物理的なAFFシステムなどを導入した場合よりも低くなるということです。さらに、Cloud Volumes ONTAPで使用されているさまざまなクラウドプロバイダが提供するストレージボリュームのパフォーマンスも継続的に向上しています。

の主なユースケースは開発とテスト作業ですが、一部のお客様は本番環境でもCloud Volumes ONTAPを使用しています。注目すべきレポートの1つは、ストレージパフォーマンスの制約を緩和するためにOracleのインメモリ機能を使用したことです。このプロセスにより、データベースサーバをホストしている仮想マシンのRAMにより多くのデータを格納できるため、ストレージのパフォーマンス要件が軽減されます。

フラッシュ

フラッシュやSSDテクノロジをOracleデータベースに使用する場合の詳しい説明は、本ドキュメントの範囲ではありません。ただし、よくある疑問と誤解の一部は取り上げる必要があります。このセクションで説明する原則は、Oracle ASMを含むすべてのプロトコルとファイルシステムに該当します。

SSDアグリゲート

SSDとフラッシュメディアをREDOログに使用することには、多くの混乱が伴います。Redoロギングのパフォーマンスを向上させるためには、データをSSDに書き込むことが必要です。SSDは直接接続されたデバイスで使用するとロギングパフォーマンスを向上させるのに役立ちますが、NetAppストレージアレイには、不揮発性、ミラーリングされたNVRAMベース、NVMEMベースのソリッドステートストレージがすでに搭載されています。Oracleデータベースが書き込み処理を実行すると、NVRAMまたはNVMEMに書き込みがジャーナルされた時点で確認応答が返されます。最終的に書き込みを受信するドライブのタイプが、書き込みパフォーマンスに直接影響することはありません。

SSDアグリゲートまたはAFFシステムを、Redoロギングなどのシーケンシャルライトのホストや一時的なデータファイルI/Oに使用しても効果はありません。ただし、AFFを選択すると、書き込みパフォーマンスが間接的に向上する場合があります。たとえば、処理するランダムI/Oが大量に発生するシステムで、回転式メディアが過負荷になると、受信する書き込みをドライブがタイムリーに処理しきれなくなり、NVMEMやNVRAMがいっぱいになることがあります。このような場合は、SSDアグリゲートまたはAFFシステムに変更することでREDOのパフォーマンスが向上しますが、これは間接的なメリットです。ランダムI/Oが適切に処理されることで書き込みパフォーマンスの問題も解決されるということです。結果として、受信する書き込みがNVMEMやNVRAMに遅延なくすべて格納できるようになり、書き込みは正常な状態に戻ります。

時には計画ミスが原因で、SSDアグリゲートを使用していてもパフォーマンスが低下することがあります。SSDは回転式メディアよりもはるかに高いパフォーマンスを提供しますが、SSDアグリゲートのデバイス数は、システム上のSASアグリゲートやSATAアグリゲートのデバイス数よりもはるかに少ない場合があります。たとえば、NetAppでは、Redoログを含む大量のシーケンシャルライトワークロードを、100本のドライブで構成される大規模なSASアグリゲートから、4~5台のデバイスで構成される小規模なSSDアグリゲートに移動することが原因で、お客様の環境で深刻なパフォーマンスの問題が発生することが確認されていますSSDはSASよりも高速かもしれませんが、無制限ではありません。

SSDアグリゲートに最も適しているのは、ランダムI/Oワークロードの処理です。インデックスはSSDへの配置に適しています。アグリゲート内のドライブ数が少なすぎるかぎり、他のタイプのI/Oに影響することはありません。ただし、以前のシステムが大幅に過負荷になっていないかぎり、パフォーマンスの向上は期待できません。

ハイブリッドアグリゲート：**Flash Pool**

NetApp Flash Poolテクノロジは、Oracleデータベースで一般に最大のパフォーマンスボトルネックとなるランダムリードのレイテンシを改善します。このテクノロジはコスト削減効果にも優れています。多くのOracleストレージシステムには、大量のスピンドルがあり、大量のランダムリードアクティビティに最小限のレイテンシで対応しています。Flash Poolを少し割り当てるだけで、多数の回転式ドライブの代わりに使用できます。

書き込みはまずNVRAMやNVMEMに送信されるので、書き込みキャッシュにFlash Poolを使用しても、書き込みパフォーマンスへの直接的な効果はありません。レイテンシの観点で言えば、I/OはデータがNVRAMやNVMEMにジャーナルされた時点で完了します。インバウンド書き込みがあとで格納されるメディアのタイプは、パフォーマンスに単独で影響することはありません。ただし、Flash Poolの書き込みキャッシュを使用すると、回転するメディアに対する負荷が軽減される場合は、書き込みパフォーマンスに間接的なメリットがあります。これにより、アレイ全体のI/Oパフォーマンスが全般的に向上する可能性があります。

Flash Pool書き込みキャッシュは、ランダムに上書きされたブロックの読み取りレイテンシも改善します。通常、データベースは書き込まれたブロックのコピーを保持するので、このプロセスはすべてのデータベースに有効なわけではありません。Oracleバッファキャッシュのサイズが大きくなり、キャッシュされる書き込み数が増えるにつれて、ブロックをドライブから再読み取りする必要がなくなる可能性があります。その場合は書き込みキャッシュを無効にして、フラッシュの貴重なスペースをランダム リード処理に残した方がよいかもしれません。

一方、同じブロックに対して繰り返されるオーバーライトをSSDレイヤにキャッシュすれば、回転式メディアの負荷が低減するというメリットが得られます。対象となるのは、Oracleのバッファ キャッシュに負荷がかかり、即座に再読み取りされるためだけにブロックがキャッシュから外されるといった状況です。

以下にNetAppの推奨事項を記載します。

- ランダムリードキャッシュとランダムライトキャッシュの両方を含むデフォルトのFlash Poolポリシーをそのまま使用します。
- 書き込みキャッシュにはメリットはないかもしれません。ほとんどのOracleデータベースにおいて、SSDのスペースが過度に使用されるほどのランダムライトは発生しませんが、デフォルトのポリシーでは必要に応じて書き込みキャッシュを使用できます。

主な例外は、データベース ワークロードに次の特性がある場合です。

- ワークロードがアグリゲートを占有し、そのためFlash Poolのキャッシングのほとんどを占めている。
- ワークロードはランダムリードのレイテンシによって制限されることがわかっています。
- 書き込みアクティビティが比較的少ない。
- Oracleのバッファ キャッシュが比較的大きい。

その場合は、Flash Poolの書き込みキャッシュポリシーをnoneに変更することを推奨します。変更すると、SSDで読み取りキャッシュに使用できるスペースが最大になります。

Flash Poolは、例えばOracle DataGuardを使用する場合など、Oracleスタンバイ データベースに対してしばしば効果があります。これは、スタンバイ データベースには普通バッファ キャッシュがなく、そのため、同じブロックに対して読み取り、更新、書き込みが何度も実行されるという厳しいI/Oパターンが発生するためで

す。Flash Poolは、集中的なオーバーライト アクティビティをSSDレイヤにキャプチャすることで、回転式メディアへの負荷を軽減します。Flash Poolなどのテクノロジが登場する以前は、レプリケーションのソースであるプライマリデータベースよりも多くの回転式ドライブがスタンバイデータベースで必要になることが珍しくありませんでした。

AFFシステム

NetApp AFFは、パフォーマンスの向上とオールフラッシュアレイ向けに調整されたデフォルトの動作によって、SSDアグリゲートの価値を拡張します。関連する各種ドキュメントは、NetApp サポートサイトで入手できます。

フラッシュの目的はIOPSだけではありません。他にも、パフォーマンスの一貫性と予測可能性、消費電力の削減、発熱量の削減、解決策の一般的な将来性などの利点があります。

オールフラッシュアレイでは、レイテンシを高めるためだけに回転式メディアを次々と導入する必要がないため、コストを削減できます。コストが大幅に減少し続けているため、デフォルトの選択肢としてAFFを選択するお客様が増えています。

ONTAP構成

ONTAPソフトウェアの設定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。2、000個の仮想データベースを含む環境のベストプラクティスは、3つの大規模なエンタープライズリソースプランニングデータベースの構成には適していない可能性があります。データ保護の要件がわずかに違うだけで、ストレージの設計に大きな影響を及ぼしかねません。本セクションでは、基本的な事項をいくつか確認していきます。詳細については、TR-4591を参照してください。設計について幅広いサポートが必要な場合は、NetAppまたはNetAppのパートナーにお問い合わせください。

RAIDレベル

NetAppストレージを構成しようとすると、RAIDレベルに関して不明点が生じることがあります。Oracleの古いガイドやOracleの設定方法に関する書籍には、RAIDミラーリングの使用には注意が必要なことや、一部のRAIDタイプの使用を避けるよう書かれているものが多く見られます。これらの資料は有効なポイントを示していますが、これらの資料は、RAID 4およびONTAPで使用されるNetApp RAID DP®およびNetApp RAID-TEC™テクノロジには適用されません。

RAID 4、RAID 5、RAID 6、RAID-DP、RAID-TECはいずれもパリティを活用して、ドライブ障害によるデータ損失を防ぎます。これらのRAIDオプションはミラーリングよりもはるかに優れたストレージ効率を発揮するのですが、大半のRAID実装では書き込み処理に影響するデメリットがあることも事実です。他のRAID実装で書き込み操作を完了するには、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です。これは、一般にRAIDペナルティと呼ばれるプロセスです。

しかしONTAPを活用すれば、RAIDペナルティは発生しません。NetApp WAFL® とRAIDレイヤが統合されているため、ペナルティは発生しません。書き込み処理がRAMで1つにまとめられ、パリティの生成も含めた完全なRAIDストライプとして用意されます。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ログなど）でパフォーマンスが妨げられたり、データファイルのランダムな書き込みで、パリティ生成によるRAIDのペナルティが発生することがありません。

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、ドライブがRAIDセット内のパートナーにリビルドされるときにドライブ障害によってデータが失われるリスクは、RAID DPセットで三重ドライブ障害が発生するリスクよりもはるかに高くなります。

容量制限

予測可能な高いパフォーマンスをストレージアレイに提供するには、メタデータとデータの構成タスク用にある程度の空きスペースが必要です。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、アグリゲートに割り当てられていないスペースや、コンスティチュエント ボリューム内の未使用のスペースを

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、この使用量が500GBのスペースを消費しているように見えます。フルプロビジョニング環境では、1TBの容量がすべて使用中と表示され、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使用されているわけではないので、本来なら、空きスペースの合計の計算に含めるべきです。

以降のセクションでは、データベースに使用するストレージ システムについてのNetAppの推奨事項を説明します。

SSDアグリゲート（AFFシステムを含む）

NetAppでは、少なくとも10%の空きスペースを確保し、未使用スペースをすべて含めることを推奨しています。この未使用スペースには、アグリゲート内またはボリューム内の空きスペースが含まれます。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も含める必要があります。

推奨する空きスペース10%は控えめな数字です。SSDアグリゲートの場合、90%以上の利用率でもパフォーマンスに影響することなくデータベース ワークロードをサポートできますが、ただし、アグリゲートの使用率が高くなると、使用率を慎重に監視しないと、スペース不足のリスクも高くなります。

HDDアグリゲート（Flash Poolアグリゲートを含む）

NetAppでは、少なくとも15%の空きスペースを推奨しています。これには、すべての未使用スペースが含まれている必要があります。この推奨事項には、アグリゲートまたはボリューム内の空きスペースと、フルプロビジョニングのために割り当てられたが実際のデータには使用されていない空きスペースが含まれます。

利用率が85%未満なら、無視できないほどの影響がパフォーマンスに及ぶことはないはずです。90%に近づくと、一部のワークロードで多少のパフォーマンス低下が目立つようになるかもしれません。95%に達すると、ほとんどのデータベース ワークロードでパフォーマンスが低下します。

Snapshotベースのバックアップ

ファイルシステムのレイアウトで最も検討すべき点は、NetApp Snapshotテクノロジの活用プランです。主に2つの方法があります。

- クラッシュ整合性のあるバックアップ
- Snapshotで保護されたホット バックアップ

データベースのクラッシュ整合性のあるバックアップを作成するには、データベースの構造全体（データファイル、Redoログ、制御ファイルなど）をある特定の時点でキャプチャすることが必要です。データベースが単一のNetApp FlexVol® フレキシブル・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格納されている場合は、整合性グループ（CG）Snapshotコピーを作成する必要があります。CG Snapshotコピーの作成には、NetApp SnapCenter® ソフトウェア、NetApp Snap Creator® フレームワーク、NetApp SnapManager® for Oracle、NetApp SnapCenter Plug-in for UNIX、ユーザが保持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ベースのバックアップは、主にpoint-of-the-backupリカバリで十分な場合に使用されます。一部の状況下ではアーカイブ ログで対応できますが、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、ホット バックアップの使用を推奨します。

Snapshotベースのホット バックアップの基本的な作成手順は次のとおりです。

1. データベースをバックアップモードにします。
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. バックアップモードを終了します。
4. コマンドを実行し `alter system archive log current` て、ログを強制的にアーカイブします。
5. アーカイブ ログをホストしているすべてのボリュームのSnapshotコピーを作成します。

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 モード時に生成された重要なアーカイブ ログを含むSnapshotコピーが1組作成されます。これらのセットは、データベースをリカバリするための2つの要件です。制御ファイルなどのファイル類も保護すると便利ですが、保護が必須となるのはデータファイルとアーカイブログだけです。

お客様によって戦略が異なる場合もありますが、これらの戦略のほとんどは、最終的にはこのセクションで概説した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。

Volume-Based 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。ただし、VBSRではボリュームのデータがすべてリバートされるので、適切でないユースケースも考えられます。たとえば、データベース全体（データファイル、Redoログ、アーカイブログを含む）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。

通常のリストアに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コピーで複製したファイルをアクティブなファイルシステムに戻すだけでリストアできます。

VBSRは、データベースが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、リカバリするデータベースのデータファイルを専用のボリュームに格納して、他のファイル タイプの影響を受けないようにしてください。SAN環境の場合は、専用のFlexVolに配置された専用のLUNに格納してください。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してください。

このように分離することで、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環境では、Oracleデータが格納された各ボリュームについて、を `percent-snapshot-space` ゼロに設定する必要があります。LUN環境ではSnapshotコピー用にスペースをリザーブしても効果はありません。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コピーに、すべてのデータの書き替えを100%吸収するのに十分な空きスペース（Snapshotリザーブを除く）がボリュームに必要になります。フラクショナル リザーブの値を100未満に設定すると、その値に応じた空きスペースが必要になりますが、この場合も、Snapshotコピーのリザーブは含まれません。つまりLUN環境の場合、Snapshotコピー用にスペースをリザーブしても無駄ということです。

一方NFS環境には、2通りのオプションがあります。

- `percent-snapshot-space` 予想されるSnapshotスペース使用量に基づいてを設定します。
- `percent-snapshot-space` ゼロに設定し、アクティブなスペースとSnapshotスペースの使用量をまとめて管理します。

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には表示されませんが、この値を設定することでスペースの利用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%の書き替えが発生した場合は、リザーブされている20%だけでなく、リザーブされていないスペースもSnapshotコピーに使用することができます。

リザーブの値を20%などに設定することには、Snapshotコピーにいつでも使用可能なスペースを確保できるという大きな利点があります。例えば、1TBのボリュームに20%のリザーブを設定すれば、データベース管理者（DBA）が格納できるデータは800GBに制限され、少なくとも200GBのスペースがSnapshotコピー用に保証されます。

`percent-snapshot-space` がゼロに設定されている場合は、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コピーを利用する1TBのボリュームがアクティブなデータとSnapshotの書き替えの間で共有されることを理解する必要があります。

エンドユーザの場合、オプション1と2のどちらを選んでも明確な違いはありません。

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、サードパーティの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可能な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、自社のスナップショットが以下の要件に沿っていることを保証しなければなりません。

- スナップショットが、Oracleが推奨するリストアおよびリカバリ処理に統合可能である。
- スナップショットが、作成時点でデータベースとのクラッシュ整合性がある。
- スナップショット内の各ファイルについて書き込み順序が保持されている

ONTAPとNetAppのOracle管理製品は、以上の要件を満たしています。

クラスタ処理-テイクオーバーとスイッチオーバー

次の操作によってデータベース操作が中断されないようにするには、ストレージ テイクオーバーとスイッチオーバーの機能を理解する必要があります。

- 通常の状態では、ある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NVMEMと呼ばれます。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にデータを保護するため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、高可用性（HA）ペアを構成する1つのノードがパートナーの処理を引き継ぎます。スイッチオーバーも基本的に同じですが、こちらはMetroCluster構成が対象で、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データベースの処理が短時間中断されることを除き、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。ネットワーキングは複雑になる可能性があり、エラーが発生しやすいです。そのため、NetAppでは、ストレージシステムを本番環境に移行する前に、データベースを使用して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ています。これ以外に、ネットワーク パスがすべて正しく設定されていることを確認する方法はありません。SAN環境では、コマンドの出力を注意深くチェックし `sanlun lun show -p` て、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ときは注意が必要です。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ます。また、代替ノードが強制的にドライブを制御します。テイクオーバーの不適切な強制は、データの損失や破損につながりかねません。この損失は、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に発生します。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てパートナーに通知し、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ます。この場合、一方のコントローラが失われ、そのコントローラ内で強制テイクオーバーが必要になります。このような状況では、コントローラ障害の前にノード間のミラーリングが失われるため、障害が発生していないコントローラに処理中の書き込みを複製することができません。そこで強制的なテイクオーバーが必要になりま

すが、その場合データが失われる可能性があります。

MetroClusterのスイッチオーバーにも同じ論理が当てはまります。通常の場合、スイッチオーバーはほぼ透過的です。ところが災害時には、セカンダリ サイトと災害発生サイトの間の接続が失われることがあります。セカンダリ サイト側から見れば、この問題は、サイト間の接続が中断されただけのことで、元のサイトでは今もデータの処理が続い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確認できなければ、強制スイッチオーバーを実行するしかありません。

NetAppでは、以下の対策を施すよう推奨しています。

- 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注意してください。強制実行は普通は必要なく、強制的な変更はデータ損失を招く恐れがあります。
- 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、データベースがシャットダウンされていること、ファイルシステムがすべてディスマウントされていること、ASMインスタンスがすべてシャットダウンされていること、論理ボリュームマネージャ（LVM）ボリュームグループがすべてvaryoffされていることを確認してください。
- 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ONTAPバージョンの『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 テクノロジですが、接続が中断すると非同期モードに切り替わります。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てデータベースI/Oが完全に停止し、データベースのサービスが停止するため、ほとんどのお客様はこの動作を推奨しています。

MetroClusterの場合、接続が再開するとアグリゲートの再同期がすぐに始まります。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。変更による差分のみが転送されます。

複数のアグリゲートにまたがって格納されているデータベースでは、災害が連続して発生した場合にデータ リカバリに追加の手順が必要になるという、ちょっとしたリスクがあります。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）アグリゲートの一部が同期されていて一部が同期されていない状態になった場合、（d）プライマリサイトが失われた場合に、さらに詳しい手順が必要です。その結果、サバイバーサイトでは、アグリゲートが相互に同期されません。このイベントが発生すると、データベースの一部が相互に同期され、リカバリなしでデータベースを起動することはできません。データベースが複数のアグリゲートにまたがって格納されている場合は、Snapshotベースのバックアップと、数多くあるツールのいずれかを活用して、この異常な事態からすばやくリカバリすることは可能かどうかを検証することを強く推奨します。

NVFAIL

データベースは大規模な内部キャッシュを保持しているため、フェイルオーバーまたはスイッチオーバーが強制的に実行されると、データベースが破損する可能性があります。強制フェイルオーバーが発生した場合、以前に確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ドライブ上のデータの状態を反映しなくなります。その結果、データが破損します。

キャッシングは、アプリケーション層またはサーバ層で行われます。たとえば、Oracleデータベース サーバは、SGA （Oracleシステムグローバル領域）内にデータをキャッシュします。SGAに格納されているブロックがアレイのブロックと一致しない可能性があるため、データが失われた操作によってデータベースが破損する可能性があります。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らかではありません。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か、LUNに基づくファイル システムはOSバッファ キャッシュにデータをキャッシュできます。このような状況でNVRAMの障害や強制テイク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ソフトウェアは、NVFAILとその関連パラメータを使用して、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。

SVMと論理インターフェイス

このセクションでは、管理に関する重要な原則をおおまかに説明します。より包括的な説明は、ご使用のONTAPバージョンに対応する『ONTAPネットワーク管理ガイド』を参照してください。データベース アーキテクチャの他の要素同様に、Storage Virtual Machine（SVM、旧称Vserver）と論理インターフェイス（LIF）の設計については、拡張性の要件とビジネス ニーズによって最適なオプションが大きく変わってきます。

LIFの戦略策定にあたっては、主に以下の事項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が十分かどうか。
- **耐障害性：**設計に単一点障害（Single Point of Failure）があるかどうか。
- **管理性：**ネットワークを無停止で拡張可能かどうか。

上記の考慮事項は、ホストからスイッチ、ストレージ システムに至る、エンドツーエンドのソリューションに該当します。

SVM

SVMはストレージの基本的な機能ユニットであるため、VMware ESX 3.5のゲストとSVMを比較すると便利です。初めてインストールしたときのESXには、ゲストOSのホスト機能やエンドユーザのアプリケーションをサポートする機能など、設定済みの機能は何もありません。仮想マシン（VM）を定義するまでは空のコンテナです。
ONTAPも同様です。インストールしただけではこのOSにデータを処理する機能はなく、SVMを定義しなければなりません。SVMの特性がデータ サービスを定義します。

お客様の中には、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処し、さらにいくつかのSVMによって次のような特殊なニーズに対応している企業があります。

- 専門チームが管理する業務上重要なデータベースを格納するSVM
- 開発グループ向けのSVM。他から独立した専用ストレージをグループで管理できるよう、管理者によって完全に制御
- 人事情報や財務レポートのデータなど、機密性の高いビジネス データを格納するSVM。管理するチームの限定が必要

マルチテナント環境では、各テナントのデータに専用SVMを割り当てることができます。SVMの最大数はクラスタ ノードあたり125個前後が推奨されますが、通常はこの最大数に達する前にLIFが最大数に達します。またマルチテナント環境は、ネットワーク セグメントを基に分離した方が、複数の専用SVMに分離するより適切です。

LIFタイプ

LIFには複数のタイプがあります。ONTAPの公式ドキュメントには、このトピックに関する完全な情報が記載されていますが、LIFの機能に関しては次のグループに分けることができます。

- クラスタ管理およびノード管理LIF：ストレージ クラスタの管理に使用するLIF。
- SVM管理LIF：SVMへのアクセスを、ONTAPのAPI（ZAPI）を通じて許可するインターフェイス。Snapshotコピーの作成やボリュームのサイズ変更などの機能に対応。NetApp 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、NFS、CIFSデータを伝送するインターフェイス。

  注：NFSトラフィックに使用するデータLIFは、ファイアウォールポリシーを `data` からに変更する `mgmt` か、HTTP、HTTPS、Secure Shell（SSH）を許可する別のポリシーに変更することで、管理にも使用できます。この変更により、NFSデータLIFと、それとは別の管理LIFの両方にアクセスするよう各ホストを設定する必要がなくなるので、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インターフェイスを設定することはできませんが、どちらもIPプロトコルを使用します。iSCSI環境の場合は、独立した管理LIFが必要です。

SAN LIFの設計

SAN環境の場合、マルチパスを使用するためLIFの設計は比較的簡単です。最新のすべてのSAN実装では、クライアントが複数のネットワーク パス経由でデータにアクセスし、アクセスに最も適したパスを1つまたは複数選択することができます。その結果、SANクライアントは利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、LIFの設計に関するパフォーマンスに対処しやすくなります。

あるパスが使用不可能になると、クライアントによって別のパスが自動で選択されます。その結果、設計がシンプルになるため、SAN LIFの管理性が向上します。これは、SAN環境の管理が常に容易であることを意味するものではありません。SANストレージには、NFSよりもはるかに複雑な要素がほかにも数多くあります。ここで言いたいのは、SAN LIFは設計が容易だということだけです。

パフォーマンス

SAN環境のLIFのパフォーマンスに関しては、帯域幅を考慮することが最も重要です。たとえば、4ノードのONTAPクラスタの各ノードに16Gb FCポートを2つずつ構成すると、ノード1つにつき最大で32Gbの帯域幅を提供できます。I/Oはポート間で自動的に分散され、すべてのI/Oが最適なパスに転送されます。

耐障害性

SAN LIFはフェイルオーバーができません。SAN LIFに障害が発生すると、クライアントのマルチパス機能によってパスの損失が検出され、別のLIFにI/Oがリダイレクトされます。

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、この移行はきわめて一般的なタスクです。SAN環境の場合は、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がパスの変更をSANに通知し、SANクライアントが自動的に再最適化するため、この機能は不要です。SAN環境でLIFの移行が必要になるのは、主に、物理ハードウェアを大幅に変更したときです。例えば、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ポートの障害が検出された場合も、LIFを未使用のポートに移行します。

設計上の推奨事項

NetAppの主な推奨事項は次のとおりです。

- パスは必要以上に作成しないでください。パスの数が多すぎると管理が全体的に複雑化し、一部のホストで、パスのフェイルオーバーによる問題が発生する恐れがあります。さらにホストによっては、SANブートなどの設定の際にパスの数が制限されるという予期せぬ事態に見舞われます。
- ストレージへのパスが4つ以上必要なLUNはほとんどありません。LUNにパスをアドバタイズするノードを3つ以上に増やしても、得られる価値には限界があります。なぜなら、LUNを所有するノードと、そのノードのHAパートナーに障害が起きると、その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からです。こうした状況では、プライマリHAペア以外のノードにパスを作成していても役に立ちません。
- 認識される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、ターゲットになる可能性のあ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マッピング機能がデフォルトです。この機能を使用すると、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。この方法を用いれば、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する必要がありません。必要最小限のノードでLUNをそれぞれ使用し、最適なパフォーマンスと耐障害性を実現できます。

  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`、コマンドを使用して追加のノードを追加し、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に新しいSANパスが作成されてLUNの移行が完了します。ただし、ホストがこの新しいパスを使用するには、パスの検出処理が必要です。

- 間接トラフィックを過度に気にする必要はありません。I/O負荷の高い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。
- 「ゾーニング」のセクションに記載されているゾーニングルールに従います。

NFS LIFの設計

パフォーマンスと耐障害性

SAN LIFのパフォーマンス測定は主に、すべてのプライマリ パスを合わせた総帯域幅を計算すれば済むことですが、NFS LIFのパフォーマンスを割り出すには、正確なネットワーク構成を詳しく確認しなければなりません。例えば、10Gbポートを2つ構成する場合、物理ポートとして構成することもできれば、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。さらに、Direct NFS（DNFS）は、現在どの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プロトコルと異なり、NFSの場合はプロトコル レイヤで耐障害性を実現しなければなりません。たとえば、LUNは常にマルチパスが有効になって設定されるため、ストレージシステムでは複数の冗長チャネルを使用できます。各チャネルはFCPを使用します。一方NFSファイルシステムは、1つのTCP / IPチャネルが使用可能かどうかに依存し、このチャネルは物理レイヤでしか保護できません。そのため、ポートフェイルオーバーやLink Aggregation Control Protocol（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がどちらもネットワーク プロトコル レイヤで提供されます。そのため、両者は互いに関連するトピックとして一緒に論じなければなりません。

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 グループに関連付けます。物理ポートを1つに集約するには、主にLACPを用います。LACPのフォールトトレランス機能は簡単です。LACPグループ内の各ポートは監視され、障害が発生すると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は、エンドポイントに合わせるためにスイッチ上で設定する必要がありません。たとえば、ONTAPではIPベースのロードバランシングを設定し、スイッチではMACベースのロードバランシングを使用できます。
- LACP接続を使用するエンドポイントは、それぞれが別々にパケット転送ポイントを選択できますが、受信に使用するポートは選択できません。この制限は、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、この到着は原因の問題ではありません。
- LACPでは、トラフィックが常に均等に分散されません。このため一般に、多数のNFSクライアントを持つ大規模環境では、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。しかし環境内のNFSは、ファイルシステム1つにつき1つのポートの帯域幅しか使用できず、アグリゲーション全体を使用することができません。
- ONTAPではラウンドロビンベースの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このポリシーを適用することはできません。たとえば、ホスト側とONTAP側でそれぞれ4つのポートをまとめてLACPトランク グループを構成しても、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側では、データの伝送に4つのポートをすべて使えますが、現在のスイッチ テクノロジでは、スイッチからホストへのデータ送信に4つのポートをすべて使用することはできません。使用できるのは1つだけです。

多数のデータベース ホストで構成された大規模環境の場合、最も一般的に用いられるのは、IPロード バランシングを使用して、適切な数の10Gb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では、クライアントが十分に存在する場合でも、すべてのポートを均等に使用できます。LACPトランキングの場合、負荷を動的に再分散することができないため、構成に含まれるクライアントの数が減るとロード バランシングが機能しなくなります。

接続が確立すると、一方向のトラフィックは1つのポートでのみ処理されます。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 スキャンを実行していて、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。耐障害性は、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実現されます。

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の動作を制御するのが、フェイルオーバー ポリシーとフェイルオーバー グループです。設定オプションは、ONTAPのバージョンが変わるごとに変更されています。具体的な詳細は、ご使用のバージョンのONTAPに対応する『ONTAPネットワーク管理ガイド』を参照してください。

ONTAP 8.2以前に関しては、以下の一般的な推奨事項に従ってください。

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、ストレージの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。このように設定すれば、間接トラフィックの生成を回避できます。
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。例えば、10Gbの1つの物理ポート上のLIFには、10Gbポート1つだけで構成されたフェイルオーバー グループを含め、4ポートのLACP LIFは、別の4ポートLACP LIFにフェイルオーバーするようにします。
4. プライマリ コントローラにフェイルオーバー ポリシーを設定します。

ONTAP 8.3では、ブロードキャスト ドメインに基づいてLIF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ポートをすべて定義し、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。一部のお客様はこの方法を使用できますが、予測性がないため、高速データベース ストレージ ネットワーク環境では限界があります。たとえば、ファイルシステムへのルーティン アクセス用の1Gbポートと、データファイルI/O用の10Gbポートを使用する環境があるとします。この2つ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ポートに移ることがあります。

NetAppでは、ONTAP 8.2によってLIFのフェイルオーバーに使用するポートを定義する方法を推奨しています。以下の推奨事項を考慮してください。

1. ユーザが定義したフェイルオーバー グループを設定します。
2. フェイルオーバー グループにSFOパートナー コントローラのポートを含め、ストレージのフェイルオーバー時にLIFがアグリゲートに従うようにします。このプロセスにより、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。例えば、10Gbの1つの物理ポート上のLIFには、10Gbポート1つだけで構成されたフェイルオーバー グループを含め、4ポートのLACP LIFは、別の4ポートLACP LIFにフェイルオーバーするようにします。これらのポートが、ブロードキャスト ドメインに定義された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。こうすることで、フェイルオーバー時にLIFがアグリゲートに従って移動します。

自動リバート

`auto-revert` 必要に応じてパラメータを設定します。ほとんどのお客様は `true`、LIFをホームポートにリバートする場合に、このパラメータをに設定します。ただし、 `false` 想定外のフェイルオーバーが発生した場合にLIFがホームポートに戻る前に調査できるように、お客様がこのパラメータをに設定することもあります。

LIFとボリュームの比率

よくある誤解の1つに、ボリュームとNFS LIFは1:1の関係にしなければならないという考えがあります。ボリュームをクラスタ内の任意の場所に移動する際に、新たなインターコネクト トラフィックの発生を抑えたいという場合はこの設定が必要ですが、すべての場合に必要なわけではありません。インタークラスタ トラフィックについては確かに考慮が必要ですが、少々の発生は問題になりません。ONTAP向けに作成され公表されているベンチ―マークの多くには、大量の間接I/Oが含まれています。

例えば、パフォーマンスが重要なデータベースの数があまり多くなく、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が1:1の戦略、つまり必要なIPアドレスの数が40個の構成が妥当かもしれません。この構成なら、任意のボリュームを任意の場所に、関連付けられているLIFと一緒に移動できます。トラフィックは常に直接伝送され、レイテンシのすべての要因をマイクロ秒レベルに押さえることができます。

一方、大規模な環境では、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。長期的に見れば、ボリュームを別のノードに移す必要が生じて、間接トラフィックが多少発生するかもしれませんが、インターコネクト スイッチのネットワーク ポートが過負荷にならないかぎり、パフォーマンスへの影響が明らかになることはありません。パフォーマンスに懸念がある場合は、ノードを追加して新しいLIFを設定したり、次回のメンテナンス期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可能です。

サービス品質（QoS）

オールフラッシュ ストレージの採用が増えた結果、データベース ワークロードも統合されました。旧式のドライブ テクノロジではIOPS機能が制限されていたため、回転するメディアに依存するストレージ アレイでは、サポートされるデータベース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。この制限が変更されました。SSDドライブの数が比較的少ないパフォーマンス機能では、最も強力なストレージ コントローラでも飽和状態になる可能性があります。つまり、回転する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ノードのHA AFF8080システムでは、レイテンシが1ミリ秒を超える前に約40万IOPSのランダムIOPSを処理できます。データベースの1%未満がこのレベルに達すると予想され、AFF8080で10、000 IOPSしか管理できないと無駄になります。

ONTAPには、IOPSと帯域幅という2種類のクオリティ オブ サービス（QoS）があります。QoS管理は、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に基づいて行われますが、IOPS QoSには直感的でない部分があります。当初、IOPSのしきい値に達したときにレイテンシが明らかに増加していたことで、困惑しているお客様もいました。これは、IOPSを制限する唯一の実行可能な方法です。論理的には、トークンシステムのように機能します。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、到着した各I/Oは処理を続行するために最初にトークンを受け取る必要があります。1秒あたり10Kを超えるトークンが消費されていない限り、遅延は発生しません。

帯域幅QoS

まず、すべてのI/Oサイズが同じではありません。データベースが大量の完全ランダムブロック読み取りを実行している可能性があり、その場合はIOPSのしきい値に達します。ただし、データベースでは、数回の大きなブロック読み取りで構成されるテーブルのフルスキャン処理も実行される場合があります。この手順は大量の帯域幅を消費しますが、IOPSは比較的少なくなります。

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のサイジングを行う必要があります

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限のQoSポリシーを適用する必要があります。その後、最小限のQoSポリシーを本番ワークロードに適用して、必要に応じて必要なIOPSの可用性が常にあるようにします。

削減率

圧縮、コンパクション、重複排除は、特定の物理ストレージに収まる論理データの量を増やすStorage Efficiency機能です。圧縮とは、簡単に言うと、データのパターンを検出してスペースが低減するようにエンコードする数学的なプロセスです。一方重複排除は、データ内で繰り返されるブロックを検出し、余計なコピーを取り除きます。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。

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、データベース圧縮の価値は限られていました。これは、ほとんどのデータベースでは、許容可能なパフォーマンスを提供するために多数のスピンドルが必要だったためです。ストレージ システムは、ドライブの数が多いことの影響で、必要以上の容量を常に備えていました。この状況はソリッドステート ストレージの普及で変わりました。優れたパフォーマンスを純粋に得るために、ドライブを過度にオーバープロビジョニングする必要がなくなったのです。ストレージ システムのドライブ スペースは、実際の容量のニーズに合わせて変更できます。

ソリッド ステート ドライブ（SSD）ではIOPSが向上するため、ほぼすべてのケースでコストを削減できますが、さらに圧縮を使用すれば、ソリッド ステート メディアの実効容量を増やして、削減効果をさらに高められます。圧縮はデータベース自体で実行できますが、Oracle環境でこの機能が使用されることはほとんどありません。組み込みの圧縮オプションでは日々変化するデータに対応することができず、高度な圧縮オプションはライセンス コストが高くつくからです。加えて、Oracleデータベース自体も安くないことを考えれば、実際のデータベース処理ではなく、データの圧縮と解凍を実行するためにCPUに高いライセンス コストを払うことには意味がありません。それよりも、圧縮処理をストレージ システムにオフロードする方が賢明です。

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与えずにOracleワークロードで徹底的にテストされます。一部のお客様から、圧縮機能を使用するとデータがキャッシュ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いう報告が寄せられています。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単位で管理するので、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8Kが使用されます。つまり、データは8K単位で圧縮されます。これは、Oracleやその他のリレーショナルデータベースで最もよく使用される8Kのブロックサイズと同じです。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。8Kの圧縮ブロックサイズよりも、32Kの圧縮ブロックサイズの方がスペース効率に優れています。ブロックサイズを大きくすると、アダプティブ圧縮でデフォルトの8Kのブロックサイズが使用されるため、削減率が若干低下します。ただし、圧縮ブロックサイズを小さくすることにも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。圧縮された32KBのデータブロックの8KBブロックを上書きするには、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。この一連の手順はストレージシステムでは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。

注： アダプティブ圧縮で使用されるブロックサイズは、最大32KBまで増やすことができます。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。アーカイブログやバックアップファイルなど、大量のデータがアレイに格納されている休止状態のファイルについては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ことがあります。ブロックサイズの拡張が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。

注意

8Kを超える圧縮ブロックサイズは、RMANバックアップで重複排除と併用しないでください。理由は、バックアップ データへのわずかな変更も32KBの圧縮領域に影響するからです。領域がずれた場合、圧縮されたデータはファイル全体にわたって変わってしまう結果となり、続けて重複排除を実行すると、重複排除エンジンは圧縮後のバックアップを別のものとみなします。RMANバックアップの重複排除が必要な場合は、8Kブロックの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。アダプティブ圧縮の方が作用するブロック サイズが小さいので、重複排除による効率化の妨げになりません。同じ理由から、ホスト側で実行する圧縮も重複排除による効率化の妨げになります。

温度に基づくストレージ効率

Temperature Sensitive Storage Efficiency（TSSE）は、ONTAP 9.8の新機能です。ブロックアクセスのヒートマップを利用して、アクセス頻度の低いブロックを特定し、圧縮して効率を高めます。

アライメント

データベース環境でアダプティブ圧縮を実行するには、圧縮ブロックのアライメントに関して考慮が必要です。これは、特定のブロックがランダムオーバーライトされるデータについてのみ考慮する必要があります。アプローチの方法としてはファイルシステム全体のアライメントによく似ています（「」のセクションを参照）。

例えば、Oracleのデータファイルに対する8KBの書き込みは、ファイルシステムの8KBの境界に合っている場合のみ圧縮されます。つまり書き込みが、ファイルの1つ目の8KBに収まり、次に2つ目の8KB、続いて3つ目の8KBにと収まっていく必要があるということです。RMANバックアップやアーカイブ ログなどは、複数のブロックにわたってデータが書き込まれるシーケンシャル書き込み処理で、すべてが圧縮されます。そのため、アライメントを考慮する必要はありません。注意が必要となるのは、データファイルのランダム オーバーライトのみです。

NFS

NFSでは、データファイルがアライメントされます。データファイルの各ブロックは、ファイルの先頭にアライメントされます。

SAN

SAN環境では、圧縮を最適化するためにはデータを8KBの境界に合わせてアライメントする必要があります。SANの場合、LUNとファイルシステムの2つの面でアライメントが必要です。LUNは、ドライブ全体として（パーティションなし）、または8Kの境界にアライメントされたパーティションで構成する必要があります。後述のOS別セクションでは、圧縮とアライメントについて構成別に詳しく解説していますので、参照してください。

注： 圧縮とフラクショナルリザベーションの相互作用については、「フラクショナルリザベーション」のセクションを参照してください。

インライン データ コンパクション

インライン データ コンパクションはONTAP 9で登場した機能で、圧縮効率を向上させます。前述のように、アダプティブ圧縮では4KのWAFLブロックに8KのI/Oが格納されるため、実現する削減率は最大でも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の影

響を受け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8K部分を更新し、再度圧縮してからドライブに書き戻すと、オーバーヘッドが発生します。

インライン データ コンパクションでは、複数の論理ブロックを物理的なブロックに格納することが可能です。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のデータによって、4KBブロック全体がその後も占有されます。インラインデータコンパクションを使用すると、1KBの圧縮データを、他の圧縮データと一緒に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。圧縮済みのデータや暗号化データは、通常それ以上は圧縮することができないため、コンパクションによるメリットはありません。初期化されたばかりのOracleデータファイルで、格納されているデータがせいぜいメタデータのブロックとゼロ ブロックの場合は、最大80:1まで圧縮できます。

重複排除

Oracleのブロックにはデータベース全体で一意のヘッダーと、ほぼ一意のトレーラが含まれています。その結果、Oracleデータベースの重複排除機能では、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。

ブロックサイズが16k以上のデータベースでは、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、データベース全体で一意のヘッダーが含まれ、最後の4KBブロックにはほぼ一意のトレーラが含まれています。内部4Kブロックは重複排除の対象となりますが、実際の重複排除による削減効果は、実際のデータではなく、初期化されたデータの重複排除によるものとほぼ同じです。

多くの競合企業は、Oracleデータベースが何回も複製されていると仮定すれば、重複排除が可能だと主張しています。この点では、NetAppの重複排除も有効ですが、ONTAPにはNetApp FlexClone®という優れたオプションがあります。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Oracleデータベースのコピーが複数作成されます。FlexCloneを使用すれば、時間をかけてデータファイルを複製し、その後重複を排除するよりも、はるかに効率的です。FlexCloneでは、重複を排除するのではなく、二重になる部分が最初から作成されません。

同じデータファイルから作成されたコピーが複数存在するまれなケースについては、重複排除のメリットが得られます。

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形態です。例えば、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。LUNを圧縮してもボリュームは100GBのままなので、実際の削減効果は目に見えません。節約したスペースをシステムのどこからでも使用できるようにするには、まずボリュームのサイズを削減することが必要です。100GBのLUNにあとから変更した結果、データの圧縮率が低下した場合は、LUNが拡張されて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、強く推奨します。その理由はシンプルです。Oracle環境では、多くの場合、空のスペース、多数のボリュームとLUN、圧縮可能なデータが含まれていることがよくあります。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予約されます。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定されます。これは、ほとんど発生することはありません。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利用できるようになり、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われます。

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、確立された運用方法に基づいてシックプロビジョニングを使用したいと考えています。

注意： 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、`sis undo` 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。volume efficiency showの出力にボリュームが表示されることはありません。この場合、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、オーバーライトの保証の動作が異なります。これによ

り、設定がオーバーサイトになるとボリュームのスペースが予期せず不足し、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では、ONTAP 9以降に次の操作を推奨しています。ONTAP 9より前のバージョンのONTAPについては、NetAppの担当者にお問い合わせください。

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。Oracleデータベースには一般に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ぼすべてのワークロードに適しています。ONTAPは、あらゆるタイプのデータと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。その結果、コストが削減されるかどうかは関係ありません。デフォルト値は、理由を完全に理解していて、逸脱することにメリットがある場合にのみ変更してください。

一般的な推奨事項

- Oracleデータベースに存在しない重複ブロックをデータをスキャンするとパフォーマンスが低下する可能性があるため、ボリュームレベルのポストプロセス重複排除を通常は使用しないでください。一方、インライン重複排除は、以前に特定された重複ブロックでのみ動作するため、問題を引き起こすことはありません。
- 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、これらの機能を使用しても節約できないため、効率性の設定を無効にする必要があります。また 、シックプロビジョニング戦略を使用する前に、「効率化とシンプロビジョニング」のセクションを参照し、効率性が完全に設定解除されていることを確認してください。
- 多数のアーカイブログを保持する場合は、32KBのブロック圧縮を使用するボリュームにアーカイブログを移動することで、効率を高めることができます。
- データファイルには、圧縮がデータベース レベルですでに有効になっている場合など、圧縮不可能な大量のデータが含まれていることがあります。 データベースには圧縮オブジェクトが含まれている場合もあれば、暗号化されている場合もあります。これらのシナリオのいずれかに該当する場合は、圧縮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ご検討ください。
- Oracle RMAN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。詳細については、セクション「IOPS QoS」を参照してください。

シンプロビジョニング

シンプロビジョニングとは、実際に使用できるスペース以上のスペースをストレージ システムに構成する手法です。この構成にはさまざまな形があり、ONTAPがOracleデータベース環境に提供する多数の機能に欠かせないテクノロジです。

Snapshotコピーを使用する場合、シンプロビジョニングが関わることがほとんどです。例えば、NetAppストレージ環境で稼働する一般的な10TBのデータベースには、30日分のSnapshotコピーが含まれています。この場合、アクティブなファイルシステムに表示されるデータは約10TB、Snapshotコピー専用のスペースが300TBとなります。合わせて310TBのストレージが、通常は約12～15TBのスペースに配置されています。アクティブなデータベースは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2～5TB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、開発に使用しました。これらのクローンを使用する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が、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

Oracle環境をシンプロビジョニングする場合は、データの変更率が予期せず上昇することがあるので注意が必要です。例えば、テーブルの再インデックス付けの際にSnapshotコピーによるスペース消費が急増したり、RMANバックアップの配置場所が不適切な場合にきわめて短時間で大量のデータが書き込まれることがあります。また、データ ファイルの拡張時にファイルシステムの空きスペースが不足して、Oracleデータベースのリカバリが困難になることもあります。

幸いなことに、これらのリスクには `volume-autogrow`、`snapshot-autodelete` ポリシーとポリシーを慎重に設定することで対処できます。名前からわかるように、この2つのオプションを使ってポリシーを作成すると、Snapshotコピーが消費するスペースを自動で削除したり、ボリュームを拡張して追加データに対応することができます。オプションは多数提供されているので、ニーズに合わせて選択してください。

これらの機能の詳細については、『ONTAP論理ストレージ管理ガイド』を参照してください。

LUNシンプロビジョニング

Oracle環境では、データファイルの作成時にファイルが初期化されてフル サイズになるため、アクティブなLUNのシンプロビジョニングはあまり有効ではありません。ファイルシステム環境でアクティブなLUNをシンプロビジョニングすると、削除または消去されたデータがファイルシステム内の未割り当てのスペースを占めるようになるため、時間の経過とともに効率が失われる可能性があります。

LVMを使用する場合は例外が1つあります。Veritas VxVMやOracle ASMなどのLVMを使用すると、基盤のLUNを複数のエクステントに分割して必要なときだけ使用できます。たとえば、サイズが2TBから始まったデータベースは、時間の経過とともに10TBに拡張される可能性があります。このデータベースは、LVMディスク グループにまとめられた、シンプロビジョニングされた10TBのLUNに配置できます。作成時に消費されるスペースは2TBにすぎず、データベースの増加に対応するためにエクステントが割り当てられたとき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れば安全です。

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LUNのスペース効率に関する動作です。このオプション `fractional-reserve` を100%に設定すると、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%書き換えることができます。

Snapshotを例に、1TBのボリュームに配置された1つの250GBのLUNにデータベースが格納されているとしましょう。Snapshotコピーを作成すると、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がスペース不足にならないようになります。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無駄です。絶対に起こらないイベントのためにスペースをリザーブ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ない状況で、スペース不足が起きないことを保証しなければならない場合は、Snapshotコピー用に100%のフラクショナル リザーブが必要となることがあります。

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一形態です。たとえば、50TBのデータベース容量を30TBに圧縮すると、20TBが削減されます。圧縮によるメリットを活かすには、削減した20TBの一部を他のデータに使用するか、あるいは50TB未満のストレージを購入する必要があります。つまり、ストレージ システムの正式な容量よりも多くのデータを格納できます。データベースから見ると、ドライブでは30TBしか使用していないにもかかわらず、50TBのデータがあります。

データベースをどれだけ圧縮できるかは常に変わる可能性があるので、実際のスペースの消費が増える結果になることもあります。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 `volume-autogrow`、監視とおよびを使用して圧縮を管理する必要があります `snapshot-autodelete`。

圧縮と重複排除については、「圧縮」と「重複排除」のセクションで詳しく説明します。

ASM再利用ユーティリティとゼロブロック検出

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、ファイルが削除されたあとにDBAがストレージアレイのスペースを再生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データが書き込まれていないため、再生プロセスが短時間で完了します。

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含むパフォーマンステストは無効です。

メモ： 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。

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は影響しません。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。ただし、圧縮機能が有効なボリューム上に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。設定の際はこの動作に困惑するかもしれませんが、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。以下の2つのシナリオを考えてください。

- フラクショナル リザーブ = 100（7.5GBを消費）
- フラクショナル リザーブ = 0（2.5GBを消費）

1つ目のシナリオでは、現在のデータ用に2.5GBのスペースと、圧縮前のデータが100%書き替えられた場合にSnapshotコピーが使用する5GBのスペースが確保されます。2つ目のシナリオでは余分にリザーブされたスペースはありません。

少し複雑かもしれませんが、実際にこうした状況に遭遇することはまずありません。圧縮には当然シンプロビジョニングが伴い、そして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。圧縮されたデータが圧縮できないデータで上書きされることはいつでも起こりえます。したがって、圧縮による削減効果を実現するためには、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。

リザーブの設定に関する推奨事項は以下のとおりです。

- `fractional-reserve` 基本的な容量監視が実施されていて `volume- autogrow` 、およびが実行されている場合は0に設定され `snapshot-autodelete`ます。
- `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に設定します。

仮想化

概要

VMware ESX、Oracle OVM、またはKernel-Based Virtual Machine（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。

仮想化のサポート ポリシーに関しては、特にVMware製品で多くの誤解があります。実際、Oracleはいっさい仮想化をサポートしないという話を耳にすることも珍しくありません。しかしこの見解は正しくなく、これを信じては仮想化の機会を逃してしまいます。Oracle Doc ID 249212.1では、Oracle環境の既知の問題について説明し、RACのサポートを指定しています。

Oracleが把握していない問題に遭遇した場合、物理ハードウェアで問題を再現するよう要求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、新しいバグが発見される可能性があるため、仮想化の使用を望まない場合があります。しかし、一般に提供されているバージョンの製品を使用しているお客さまで、仮想化に際してこのような状況が問題になったことはありません。

ストレージ提供

データベースの仮想化を検討する場合、ビジネス ニーズを基にストレージを決めることが必要です。この原則はすべてのIT意思決定に当てはまりますが、プロジェクトの規模と範囲が大きく異なるため、仮想化にとって特に重要です。

ストレージ プレゼンテーションには、次の4つの基本的なオプションがあります：

- 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します。
- NFSファイルシステムを、仮想マシン ディスク（VMDK）ではなくVMでマウントします。
- ハイパーバイザー データストア

原則として、Oracleファイルにデータストアを使用することは避けてください。これには次のように多くの理由があります。

- **透明性：**VMが自身のファイルシステムを所有していると、データベース管理者やシステム管理者がデータについてファイルシステムのソースを特定しやすくなります。
- **パフォーマンス：**テスト結果から、すべてのI/Oをハイパーバイザー データストア経由でチャネリングすると、パフォーマンスに影響することがわかっています。
- **管理性：**VMが自身のファイルシステムを所有していると、ハイパーバイザー レイヤを使用するかどうかが管理性に影響します。ハイパーバイザー レイヤを使用していれば、サイトに仮想環境と非仮想環境が混在していても、同じ手順でサイト全体のプロビジョニング、監視、データ保護などを実行できます。
- **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と、ストレージ スタック全体がVM上に存在するため、安定した高パフォーマンスが実現し、問題のトラブルシューティングも一段と簡単になります。ハイパーバイザーは、FCやIPのフレームを転送する役目だけを担います。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、および考えられるバグが新たに追加されるため、構成が複雑になります。
- **移動性：**VMが自身のファイルシステムを所有していると、Oracle環境を移動するプロセスはずっとシンプルになり、ファイルシステムを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すべてにおいてきわめて困難になります。
- **Snapshotの有効化：**帯域幅が比較的限られているため、仮想環境でのバックアップが問題になることがあります。たとえば、多数の仮想データベースで日々必要とされるパフォーマンスを満たすには、4ポートの10GbEトランクで十分だったとします。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となると、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ため、このトランクでは不十分です。

  VM所有のファイルシステムを使用すると、Snapshotベースのバックアップとリストアを簡単に活用できます。VM所有のファイルシステムはストレージ システムにバックアップ処理をオフロードするため、バックアップウィンドウの帯域幅とCPUの要件を満たすためだけに、本来なら不要なハイパーバイザー構成を構築する必要はありません。

以下にNetAppの推奨事項を記載します。

- パフォーマンスと管理性を最適化するためには、Oracleデータをデータストアに配置することは避けてください。ゲスト所有のファイルシステム（ゲストまたはRDMで管理されるNFSまたはiSCSIファイルシステムなど）を使用します。

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。データストアが使用されている場合、準仮想化SCSIドライバは必須です。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異なり、ゲストがより緊密にハイパーバイザーと統合されます。エミュレートされたドライバを使用した場合、ハイパーバイザーは物理ハードウェアの動作の模倣により多くのCPU時間を消費します。

ほとんどのデータベースのパフォーマンスはストレージによって制限されます。そのため、ネットワークまたはSCSIドライバによって発生する余分な遅延が顕著になります。NetAppのカスタマー サポートに寄せられるパフォーマンスに関する苦情の多くは、準仮想ドライバをインストールすることで解決されています。あるお客様のPOCでは、データベースにESXを使用したところ、同じハードウェアをベア メタルとして使用した場合よりもパフォーマンスが向上しました。テストには大量のI/Oを使用しましたが、パフォーマンスに違いが出たのは、ESX準仮想ネットワーク ドライバを使用したことが理由です。

以下にNetAppの推奨事項を記載します。

- 準仮想化ネットワーク ドライバとSCSIドライバを常に使用してください。

RAMのオーバーコミット

RAMのオーバーコミットとは、物理ハードウェアに存在する容量よりも多くの仮想RAMを、様々なホストに構成することです。これは、パフォーマンスに予期せぬ問題を引き起こす可能性があります。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。ブロックがスワップ アウトされると、パフォーマンスがきわめて不安定になります。

以下にNetAppの推奨事項を記載します。

- ハイパーバイザーは、Oracle SGAのブロックがスワップ アウトされないように設定します。

一般的なOracle構成

次のパラメータはすべての構成に適用できます。

filesystemio_options

Oracleの初期化パラメータは、`filesystemio_options` 非同期I/OとダイレクトI/Oの使用を制御します。一般的な言説と異なり、非同期I/OとダイレクトI/Oはお互いに相反するものではありません。このパラメータを正しく設定しているお客様はあまり多くなく、誤った設定が、パフォーマンスに関する多くの問題の直接的原因になっています。

非同期I/Oとは、Oracle I/Oの並行処理が可能ということです。さまざまなOSで非同期I/Oを使用できるようになる以前、ユーザは大量のdbwriterプロセスを設定し、サーバの処理設定を変更していました。非同期I/Oでは、OSがデータベース ソフトウェアに代わって、効率性に優れた並行処理でI/Oを実行します。この処理によってデータがリスクにさらされることはありません。また、OracleのRedoログ作成などの重要な処理は、引き続き同期I/Oで実行されます。

ダイレクトI/OはOSのバッファ キャッシュを迂回します。UNIXシステムのI/Oは通常、OSのバッファ キャッシュを経由します。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るので、ほぼすべてのケースで、ダイレクトI/Oを有効にしてサーバのRAMをSGAに割り当てた方が、OSのバッファ キャッシュを使用するよりも適しています。OracleのSGAはメモリをより効率よく使用します。またI/OがOSのバッファを経由した場合、追加の処理が必要になってレイテンシが増大します。このレイテンシの増大は、低レイテンシが重要な要件である大量の書き込みI/Oで問題となります。

のオプションは `filesystemio_options` 次のとおりです。

- **Async：**OracleからOSにI/O要求が送信されて処理されます。OracleはI/Oの完了を待たずに別の作業を実行できるので、I/Oの並行処理が促進されます。
- **directio：**Oracleは、ホストOSのキャッシュ経由でI/Oをルーティングする代わりに、物理ファイルに対して直接I/Oを実行します。
- **none：**Oracleは同期I/OとバッファI/Oを使用します。この設定では、サーバ処理を共有するか占有するかの選択と、データベースへの書き込み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を両方使用します。

ほとんどの場合、の使用 `setall` は最適ですが、次の問題を考慮してください。

- 一部のお客様から、特に、以前のRed Hat Enterprise Linux 4（RHEL4）リリースで、過去に非同期I/Oに関する問題が報告されていました。ただし現在この問題が報告されることはなく、非同期I/Oは最新のすべてのOSで安定して機能しています。
- データベースがバッファI/Oを使用していた場合は、ダイレクトI/Oに切り替えることで、SGAサイズの変更が必要になる場合もあります。バッファI/Oを無効にすると、ホストOSのキャッシュがデーベースにもたらすパフォーマンス上のメリットがなくなります。ただしこの問題は、RAMをSGAに戻すと解決され、最終的にはI/Oパフォーマンスの向上という結果が得られます。
- RAMは、OSのバッファ キャッシュよりもOracle SGAに使用した方がよい場合がほとんどですが、選択が難しい場合もあります。たとえば、データベース サーバのSGAのサイズがごく小さく、断続的にアクティブになるOracleインスタンスが多数ある場合は、バッファI/Oの使用が好ましいかもしれません。そうすることで、使用されずに空いているOSのRAMを、実行中のすべてのデータベース インスタンスに柔軟に使用できます。これはきわめて稀なケースですが、一部のお客様では実際にこのような状況が報告されています。

  メモ： `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。

以下にNetAppの推奨事項を記載します。

- `filesystemio_options` に設定します `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。

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中に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。影響するのはシーケンシャルI/Oのみです。

Oracle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。設定しないでおけば、データベース ソフトウェアが自動で最適な値を設定します。具体的には、このパラメータはI/Oサイズが1MBになるような値に設定されます。たとえば、8KBブロックを1MBで読み取るには128個のブロックを読み取る必要があるため、デフォルト値は128に設定されます。

データベースのパフォーマンスに関して、お客様のサイトに見られる問題のほとんどには、このパラメータが正しく設定されていないことが関係しています。Oracleのバージョン8と9では、ユーザがこの値を変更する余地が残されていたため`init.ora` データベースがOracle 10以降にアップグレードされているため、このパラメータがファイルに含まれていることがあります。このパラメータが8や16に設定されていると、デフォルト値の128と比べてシーケンシャルI/Oのパフォーマンスに多大な影響を及ぼします。

以下にNetAppの推奨事項を記載します。

- `db_file_multiblock_read_count` パラメータを `init.ora` ファイルに含めることはできません。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のスループットが損なわれることが明らかなケースは少なくありません。

Redoブロックサイズ

Oracleは、512バイトまたは4KBどちらかのRedoブロック サイズに対応しています。デフォルトは512バイトです。Redo処理の際に書き込まれるデータの量が最小限に抑えられるため、これが最良のオプションです。ただし、ログが頻繁に記録される場合は、4KBのサイズを使用することでパフォーマンスが向上する可能性があります。たとえば、50MBpsでRedoログが作成される単一のデータベースの場合、Redoブロック サイズが大きい方が効率的な可能性があります。ストレージ システムが多数のデータベースをサポートしていて、Redoログの合計が大容量になるケースでは、Redoブロック サイズを4KBにした方がメリットがあるかもしれません。これは、4KBブロックの一部のみを更新する必要がある場合に非効率的な部分I/O処理が行われなくなるためです。

すべてのI/O処理がRedoログのブロック サイズ単位で実行されるわけではありません。ロギング速度が非常に高い場合、一般にデータベースは複数のREDOブロックで構成される大規模なI/O処理を実行します。この場合、これらのRedoブロックの実際のサイズがロギングの効率に影響することは通常ありま

せん。以下にNetAppの推奨事項を記載します。

- デフォルトのブロック サイズは、特定のアプリケーションに関して文書化された要件がある場合や、NetAppまたはOracleのカスタマー サポートから推奨された場合にのみ変更してください。

チェックサムとデータ整合性

NetAppには、データベースのデータ整合性をどうやって確保すべきかという質問がよく寄せられます。この質問は、Oracle RMANのストリーミングバックアップを使い慣れているお客様がSnapshotベースのバックアップに移行する場合によくあります。RMANには、バックアップ処理の際に整合性チェックを実行する機能が備わっています。この機能に意味がないわけではありませんが、主にメリットがあるのは、データベースが最新のストレージ アレイを使用していない場合です。Oracleデータベースに物理ドライブを使用している場合、ドライブが古くなって最終的に破損することはほぼ確実です。この破損には、実際のストレージアレイでアレイベースのチェックサムが使用されています。

本物のストレージ アレイでは、複数のレベルでチェックサムが実行され、データ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、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。FCP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にはRAIDとチェックサムによる保護があるため、データの破損が避けられないとしても、ほとんどのエンタープライズ アレイ同様、検出されて修正されます。もっとも多いのはドライブ全体の破損で、この場合はRAIDのリビルドが要求され、データベースの整合性が維持されます。ONTAPがチェックサムエラーを検出することもあります。これは、ドライブ上のデータが破損していることを意味します。ドライブが故障し、RAIDのリビルドが開始されます。この場合も、データの整合性には影響はありません。

OracleのデータファイルとRedoログ アーキテクチャも、最悪の状況下でも可能な限り高水準のデータ整合性を提供するよう設計されています。最も基本的なレベルで言えば、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で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らないかぎり、 データは維持されます。データ整合性のチェック レベルは調整可能で、また、書き込みを確認するようOracleを設定することもできるので、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、まれにリカバリ不能な状況が発生した場合は、破損が即座に検出されます。

Oracleデータベースを使用しているNetAppのお客様は、一度Snapshot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を使わなくなることがほとんどです。RMANを使用できるオプションとしては、SMOによるブロックレベルのリカバリがまだあるにはありますが、日常レベルで言えば、RMANやNetBackupなどの製品は、1カ月または四半期に一度アーカイブ コピーを作成するのに時々使用される程度です。

一部のお客様は、を dbv 定期的に実行して既存のデータベースの整合性チェックを実行することを選択しています。これは不要なI/O負荷を生成するので、NetAppでは推奨していません。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 問題が検出される可能性はほぼゼロです。このユーティリティを使用すると、ネットワークおよびストレージシステムに高いシーケンシャルI/O負荷が発生します。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、を実行する理由はありません dbv。

一般的なSAN構成

ゾーニング

FCゾーンに複数のイニシエータを配置することは絶対に避けてください。イニシエータを複数配置すると、最初は問題なく動作しているように見えても、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や安定性を妨げます。

マルチターゲットゾーンは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。例えば、NetApp製とEMC製のストレージ アレイのターゲット ポートを、どちらも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ことは、問題が生じる可能性をさらに高めます。

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関するI/Oを最適化することです。NetAppシステムの場合、ストレージのまとまりは4KB単位なので、Oracleデータファイルの8KBブロックを、4KBブロック2つに正確に揃えます。LUNの構成エラーのためにアライメントがどちらかの方向に1KBずれると、8KBのOracleブロック1つが、4KBのストレージ ブロック2つではなく3つにまたがってしまいます。この状態は、レイテンシの増大やストレージ システム内で実行されるI/Oの増大につながります。

LUNのアライメントは、論理ボリュームマネージャを使用しない場合にのみ考慮する必要があります。現実的な問題として、この考慮事項は、LinuxとSolarisが主要な関心事であることを意味します。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。このレイアウトは正しい位置合わせです。問題が起きるのはパーティションありの場合で、パーティションによって、OSがLUNを使用し始める場所がずれます。オフセットを4KB単位でシフトすると、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 `fdisk -u` 、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。具体的には、パーティションが8の倍数の512バイトのセクター（4KB）で始まるということです。

圧縮ブロックのアライメントについては、「圧縮と重複排除」のセクションも参照してください。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。

Solaris環境のアライメントは、より複雑です。詳細については、該当するHost Utilitiesのガイド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、アライメントを適切に行うには、より注意が必要です。Solaris x86のパーティション スライスは、通常、標準のマスター ブート レコードのパーティション テーブルの上にあります。

LUNのミスアライメントの警告

OracleのRedoロギングは通常アライメントされていないI/Oを生成するため、ONTAPでLUNがミスアライメントされているという警告が誤って発行されることがあります。Oracle Redoロギングは、Redoログ ファイルのシーケンシャル オーバーライトを様々なサイズの書き込みで実行します。4KBの境界にアラインしていないログの書き込み処理があっても、次のログ書き込み処理が4KBブロックを埋めるため、一般にパフォーマンスの問題が起きることはありません。その結果、一部の4KBブロック内のデータが2つの別 々 の処理で書き込まれていたとしても、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。

`sio dd` 定義したブロックサイズでI/Oを生成できる、やなどのユーティリティを使用してアライメントを確認します。ストレージシステムのI/Oアライメント統計は `stats`、コマンドを使用して表示できます。詳細については、「付録B：WAFLアライメントの検証」を参照してください。

LUNのサイジング

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スピンドル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すべての機能を利用するため、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を使用することもあります。例えば、データベースの基盤になるASMディスクグループを2つの1TB LUNで構築した場合、そのASMディスクグループは1TB単位で拡張しなければなりません。8個の500GB LUNでASMディスクグループを構築した方が、ディスクグループの増分単位が小さくなって便利だと思われるかもしれませんが、

汎用性の高い標準的なサイズでLUNを構成することは、管理が複雑になる恐れがあるので推奨できません。例えば、標準サイズの100GB LUNは、1～2TB規模のデータベースには使いやすいかもしれませんが、20TBのデータベースの場合はLUNが200個必要になります。この違いは、サーバのリブートに時間がかかり、さまざまなUIで管理するオブジェクトが増え、SMOなどの製品では多数のオブジェクトに対して検出を実行する必要があることを意味します。LUNのサイズを大きくして数を減らせば、こうした問題は避けられます。

注：

- LUNの数は、サイズ以上に重要です。
- LUNのサイズは、必要なLUNの数で決まることがほとんどです。
- 必要以上の数のLUNを作成することは避けてください。

LUNのサイズ変更とLVMベースのサイズ変更

SANベースのファイルシステムが最大容量に達した場合、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、グループ内の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、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を使用すると、複数のLUNを1つの仮想ストレージ プールにまとめることができます。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。さらに、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という利点もあります。透過的な移行は、一般に、論理ボリューム マネージャを使用して、論理ボリュームの基盤であ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。

LUNの数

LUNのサイズと異なり、LUNの数はパフォーマンスに大きく影響します。Oracleデータベースのパフォーマンスは、SCSIレイヤを介して並列I/Oを実行する能力によって左右されます。そのため、LUNは1つよりも2つ使用した方がパフォーマンスが向上します。Veritas VxVMやLinux LVM2、Oracle ASMなどのLVMを使用すると、並行処理を簡単に促進できます。

NetAppをご利用のお客様は、LUNの数を16以上に増やすことで得られるメリットはごくわずかですが、ランダムI/Oが多い100% SSD環境のテストでは、最大64個のLUNがさらに向上したことが実証されています。

以下にNetAppの推奨事項を記載します。

- 一般に、データファイルI/Oをサポートするには、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。

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。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、その1MBのデータを1つのドライブから読み取るには、1MBのデータがゆっくり転送されるため、大量のディスクヘッドが必要になります。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 並行して問題できます。これにより、1MBの転送を完了するの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。

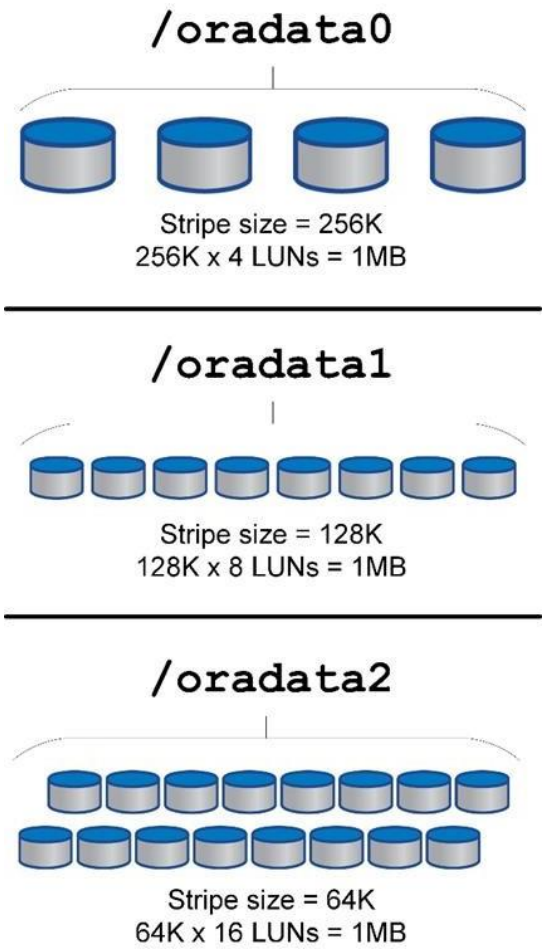
Oracle ASMなどのLVMはデフォルトでストライプされますが、ネイティブOS LVMはストライプされません。その中には、複数のLUNを連結されたデバイスとして結合するものもあります。そのため、データファイルは1つのLUNデバイスにしか存在しません。これにより、ホットスポットが発生します。他のLVM実装では、デフォルトで分散エクステントが使用されます。これはストライピングのようなものですが、粗いです。ボリュームグループ内のLUNはエクステントと呼ばれる大きな単位にスライスされ、通常は数メガバイト単位で測定されます。データベースで使用される論理ボリュームは、それらのエクステントに分散されます。そのため、データ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Oracle I/Oは、ほとんどの場合（a）Oracleブロックサイズの単位または（b）1MBのいずれかです。RedoロギングのI/Oのサイズはさまざまですが、ボリュームはデータベースI/Oよりもはるかに小さく、ストライピングによるメリットを得ることはほとんどありません。

ストライプ構成の主な目的の1つは、単一ファイルI/Oを単一のユニットとして実行できるようにすることです。もう1つの目的は、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で均等に並列処理することです。この処理を実行するには、ストライプ・サイズをデータベース・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1に、考えられる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。

図1）ストライプサイズと幅の調整



データファイルのブロックサイズ

OSの中には、ファイルシステムのブロック サイズを選択できるものがあります。ファイルシステムがサポートするデータファイルに関しては、圧縮を使用する場合、ブロック サイズを8KBにしてください。圧縮が不要な場合、ブロック サイズは8KBと4KBのいずれでも構いません。

OSの中には、ファイルシステムのブロック サイズを選択できるものがあります。データファイルをサポートしているファイルシステムに関しては、ブロック サイズを4KBにしてください。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がNetAppの推奨事項を基に適切にアライメントされていても、ファイルのI/Oは適切にアライメントされないかもしれません。その場合は、パフォーマンスに深刻な問題が発生します。

Redoブロックサイズ

Redoログをサポートしているファイルシステムには、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。このサポートには、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 サイズを試す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数分単位のトランザクションで使用しているデータベースについて、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バイト単位の変更を4KBの１つ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すが、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。

以下にNetAppの推奨事項を記載します。

- Redoブロックサイズは、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アドバイスがあった場合、または公式の製品ドキュメントに基づいて変更された場合を除き、変更しないでください。

一般的なNFS設定

NFSクライアント

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標準のNFSドライバ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NFS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が適切に設定されている場合、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、NFSファイル操作をNFSサーバ上で直接実行することです。そのために必要となるのは、Oracle Disk Managerライブラリの変更だけです。処理の手順はOracleのドキュメントに記載されています。

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のオプションがあります。例えば、2つの10Gbインターフェイスをバインドして、20Gbの帯域幅を提供することが可能です。一方のインターフェイスに障害が発生すると、もう一方のインターフェイスでI/Oが再送信されます。全体的な処理はFCマルチパスに似ています。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に記載のパッチがすべてインストールされていることが重要です。パッチをインストールできない場合は、環境を検証して、Oracleのドキュメント1495104.1に記されているバグによって問題が発生し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。

注意

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。これらの解決策には、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、以降の検索でも同じIPアドレスが使用される必要があります。変更を行うと、Oracleデータベースがクラッシュし、データの不整合が発生する可能性があります。

Direct NFSとホストファイルシステムへのアクセス

DNFSを使用すると、ホストにマウントされている参照可能なファイルシステムに依存するアプリケーションやユーザのアクティビティで原因の問題が発生することがあります。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ことが原因で、問題が発生する可能性があります。DNFSクライアントがファイルの作成、削除、修正を行ってもOSは認識できないことが原因です。

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、ファイルやディレクトリの属性のキャッシュが有効になり、結果としてディレクトリの内容もキャッシュされます。そのため、DNFSがファイルを作成すると、OSがディレクトリの内容を再度読み取ってファイルが参照可能になるまでに、若干の遅延が生じます。この遅延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。問題がある場合は、マウントオプションをOracle RACに関する推奨事項に変更して、問題に対処します。ただしこの変更によってホストのキャッシングがすべて無効になります。

マウント オプションは、（a）DNFSを使用していて、（b）問題がファイルが参照可能になるまでの遅延に起因している場合にのみ変更してください。DNFSを使用していない場合に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。

メモ： `nosharecache` 異常な結果が発生する可能性があるLinux固有のDNFS問題については、「Linux NFSのマウントオプション」のに関する注を参照してください。

NFSのバージョン

オペレーティングシステムのNFSクライアントがNetAppでサポートされている必要があり、DNFSを使用する場合は、NFSクライアントも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でサポートされています。DNFSは、Oracleのすべてのバージョンで NFSv3で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のDNFSサポートには、 Oracle 12.2.0.2以降が必要です。
- NFSv4.1では、特定のOSサポートが必要です。サポート対象のOSについては、NetApp Interoperability Matrix Tool（IMT）を参照してください。NFSv4.1のDNFSサポートには、Oracleバージョン19.3.0.0以降が必要です。

注： NFSv3およびNFSv4のNetAppサポートマトリックスには、特定のオペレーティングシステムは含まれていません。RFCに準拠するすべてのOSがサポートされます。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、一致する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。

ONTAP NFS転送サイズ

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。Oracleデータベースのランダム I/Oで使用されるブロックサイズは、64Kの最大サイズよりもはるかに小さくなります。ラージブロック I/Oは並列化されるため、最大64Kも制限事項ではありません。

一部のワークロードでは、最大64Kに制限があります。フルテーブルスキャンやRMANバックアップなどのシングルスレッド処理は、データベースで実行するI/Oが少なくても大容量であれば、より高速かつ効率的に実行できます。Oracleワークロードを実行するONTAPに最適なI/O処理サイズは256Kです。以下のオペレーティングシステムのNFSマウントオプションは、それに応じて64Kから256Kに更新されています。

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。

```
EcoSystems-A200-A::> 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

EcoSystems-A200-A::*> nfs server modify -vserver jfsCloud3 -tcp-max-xfer-size 262144
```

```
EcoSystems-A200-A::*>
```

注意

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。

インストールとパッチ適用

に次のマウントオプションがある `ORACLE_HOME` と、ホストのキャッシュが無効になります。

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

これは、Oracleソフトウェアのインストールとパッチ適用の速度に深刻な悪影響をもたらす恐れがあります。お客様の多くは、Oracleバイナリのインストール時やパッチ適用時に、このマウント オプションを一時的に削除します。`ORACLE_HOME` 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をアクティブに使用しているプロセスが他に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。

ADRとNFS

一部のお客様から、 `ADR` 場所にある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。この問題は一般に、高いパフォーマンスを必要とするデータが大量に蓄積されて初めて発生します。I/Oが過剰に発生する理由は不明ですが、どうやら、Oracleが対象のディレクトリを何度もスキャンして変更しようとしていることに原因があるようです。

`noac` および `actimeo=0` /またはマウントオプションを削除すると、ホストOSのキャッシュが実行され、ストレージのI/Oレベルが低下します。

以下にNetAppの推奨事項を記載します。

- `ADR noac actimeo=0` パフォーマンスに問題が生じる可能性があるため、またはが設定されたファイルシステムにはデータを配置しないでください。`ADR` 必要に応じて、データを別のマウントポイントに分離します。

nfs-rootonlyおよびmount-rootonly

ONTAPには、というNFSオプションがあります `nfs-rootonly` 。このオプションは、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を制御します。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、プロセスはrootとして実行されるため、の値を変更する必要はありません `nfs-rootonly`。接続は低ポートから行われます。

`mount-rootonly` [Only環境NFSv3]オプションを選択します。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 disabled` dNFSが接続を完了するには、パラメータをに設定する必要があります。

`nfs-rootonly` を有効にすると、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。

```
EcoSystems-A200-A::> nfs server modify -nfs-rootonly disabled

EcoSystems-A200-A::>
```

注： まれに `nfs-rootonly`、との両方を `mount-rootonly` に変更しなければならない場合があります `disabled`。サーバが多数のTCP接続を管理している場合は、1024未満のポートが使用できず、OSがより高いポートを使用することを余儀なくされる可能性があります。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。このアクションは、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。この再マッピングは、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、データの保護に役立ちます。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 `allow-suid` 、ルールにとの両方が含まれ、 `superuser` システム認証を使用したNFSクライアントへのアクセスが許可されます（ルート）。

```
EcoSystems-A200-A::> export-policy rule show -vserver jfsCloud3 -policyname jfs0 -fields allow-
suid,superuser
vserver   policyname ruleindex superuser allow-suid
--------- ---------- --------- --------- ----------
jfsCloud3 jfs0       1         sys       true
```

NFSv3 TCPスロットテーブル

TCPスロット テーブルは、ホスト バス アダプタ（HBA）キュー深度に相当するNFSv3環境の機能で、同時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実現するには少なすぎます。その一方で、新しいLinuxカーネルでは、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まで自動で引き上げることが可能なため、逆の問題が起きています。

パフォーマンスを最適化し、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のパラメータを調整する必要があります。

`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
```

sunrpc.tcp_slot_table_entriesはすべてのLinuxシステムで表示されますが、sunrpc.tcp_max_slot_table_entriesが表示されるのは一部のシステムのみです。どちらも128に設定します。

注意

- 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ことがあります。
- 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。

NFSv4 / 4.1構成

Oracleデータベースに関しては、NFSv3とNFSv4の違いはほとんどありません。Oracle I/Oは単純な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。

NFSv4の機能が必要な場合はNetApp、NFSv4.1の使用を推奨します。NFSv4.1では、一部のエッジの場合の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。Oracleデータベースでは、NFSv4.2はまだサポートされていません。

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。OSの設定に関するガイダンスなど、ONTAPを使用したNFSv4設定の詳細については、TR-4067『NFS on NetApp ONTAP Best Practices』を参照してください。このTRの以降のセクションでは、NFSv4でOracleを使用するための基本的な要件の一部を説明します。

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、データベースからは、 oradism 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。DBAは、chown chmod 実際の問題がドメイン名であったときに、Oracleバイナリのアクセス許可が破損し、またはコマンドを実行していると誤って判断することがあります。

ONTAP SVMでNFSv4ドメイン名が設定されます。

```
EcoSystems-A200-A::> nfs server show -fields v4-id-domain
vserver   v4-id-domain
------------ -----------------
jfsCloud3 jfs.lab
```

ホストのNFSv4ドメイン名は、 /etc/idmap.cfg

```
[root@jfs0 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 = jfs.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/var/log/messagesます。

```
Apr 12 11:43:08 jfs0 nfsidmap[16298]: nss_getpwnam: name 'root@jfs.lab' does not map into domain
'default.com'
```

Oracleバイナリには、rootが所有するsetuidビットのファイルが含まれます。つまり、NFSv4ドメイン名が一致していないと、Oracle dNFSの起動に失敗し、oradism$ORACLE_HOME/bin ディレクトリにあるという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。次のように表示されます。

```
[root@jfs0 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ドメインマッピングに問題がある可能性があります。

```
[root@jfs0 bin]# ls -l oradism
-rwsr-x--- 1 nobody oinstall 147848 Apr 17 2019 oradism
```

この問題を解決するには、`/etc/idmap.cfg` ファイルをONTAPのv4-id-domain設定と照合し、整合性があることを確認します。そうでない場合は、必要な変更を行い、を実行して、`nfsidmap -c`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。`chown root` NFSドメインの設定が修正される前に、ユーザがこのファイルに対してを実行しようとした場合は、を `chown root` 再度実行しなければならないことがあります。

イーサネット構成

Oracleデータベースソフトウェアのインストールに必要なTCP/IP設定は、すべてのNFSまたはiSCSIストレージリソースに優れたパフォーマンスを提供するのに十分です。NetAppでは、ネットワークアダプタメーカーからの特定の推奨事項を実装した後、10Gb環境でパフォーマンスが向上することがあります。

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元に対し、データの送信を一時停止するよう要求できます。この要求は、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実行されます。かつては、送信元にデータ転送の中止を要求した方が、バッファがフルになってから受信側でパケットを廃棄するよりもシステムへの影響を抑えることができました。現在OSで使用されているTCPスタックでは、この手順は当てはまりません。実際フロー制御では、問題が解決されるよりも、問題が起きる例の方が多く見られます。

特に近年は、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 フロー制御が物理層で動作することに理由があります。すべてのデータベース サーバがイーサネット フロー制御要求をストレージ システムに送信できるようネットワークが構成されていたら、接続しているクライアントすべてに対するI/Oが停止する結果になります。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ますます増えていることから、そのうちのいくつかのクライアントがフロー制御要求を送信する確率は高まっています。OSの仮想化を幅広く導入しているお客様のサイトでは、頻繁にこの問題が起きるようになりました。

NetAppシステムに搭載されているNICでは、フロー制御要求を受信しないようにしてください。その方法は、ネットワーク スイッチのメーカーによって異なります。通常、イーサネットスイッチのフロー制御は `receive desired` またはに設定できます `receive on`。つまり、フロー制御要求はストレージコントローラに転送されません。ただし、ストレージ コントローラのネットワーク接続によっては、フロー制御の無効化が許可されない場合があります。このような場合は、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。このような要求には'データベース・サーバ自体のNIC構成を変更するか'データベース・サーバが接続されているスイッチ・ポートを変更することによって完了した要求も含まれます

以下にNetAppの推奨事項を記載します。

- NetAppのストレージ 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します。この手順は通常、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。

ジャンボ フレーム

1Gbネットワークにジャンボ 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低減してパフォーマンスがいくらか向上しますが、通常、顕著な効果は得られません。こうした状況でも、可能であればジャンボ フレームを実装するようNetAppでは推奨しています。それは、パフォーマンスの向上と将来のニーズに応えるソリューションの2つを同時に実現できるからです。

10Gbネットワークにはジャンボ フレームの使用がほぼ必須条件です。ほとんどの10Gb環境の場合、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最大値に達してしまいます。ジャンボフレームを使用すると、データベース サーバ、NIC、ストレージ システムが処理する大容量のパケットの数が減るため、TCP / IP処理が効率化されます。NICによって差はありますが、パフォーマンスは大幅に向上します。

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、Maximum Transmission Unit（MTU；最大伝送ユニット）サイズがエンドツーエンドで一致してい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9,216、NetAppコントローラは9,000にそれぞれ設定されており、クライアントは9,000と1,514が混在しています。MTU9,000に対応できるクライアントにはジャンボ フレームを使用でき、1,514しか対応できないクライアントはより低い値でネゴシエートできます。

スイッチだけで接続された環境では、この設定で問題が生じることはまずありません。ただしルーティングされた環境では、ルータによってジャンボ フレームが断片化されないよう注意が必要です。

以下にNetAppの推奨事項を記載します。

- 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。
- ジャンボ フレームは10GbEのパフォーマンスを最大化するために必要です。

TCPパラメータ

TCPタイムスタンプ、 選択的確認（SACK）、TCPウィンドウの拡張の3つは、設定ミスの多いパラメータです。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機能が今よりもはるかに低く、TCP処理のオーバーヘッドをできるだけ軽減することに効果があった何年も前であれば、この推奨事項にもいくらかメリットがありました。

しかし、最新のOSでは、これらのTCP機能を無効にしても通常は何もメリットもないだけでなく、パフォーマンスが低下する可能性もあります。特に仮想ネットワーク環境では、この3つの機能が有効でないと、パケット損失やネットワーク品質の変化に効率よく対処できず、パフォーマンスが低下する可能性が高まります。

以下にNetAppの推奨事項を示します。

- ホストのTCPタイムスタンプ、SACK、TCPウィンドウ拡張を有効にします。

クラスタリング

Oracle Real Application Clusters

このセクションの内容が当てはまるのは、Oracle 10.2.0.2以降です。それより前のバージョンのOracleについては、本ドキュメントでOracleのドキュメントID 294430.1を参照して、最適な設定を確認してください。

disktimeout

プライマリ ストレージに関連するRACパラメータは`disktimeout`です。このパラメータは、VotingファイルI/Oがそれまでに完了しなければならないタイムアウトを制御します。`disktimeout` パラメータの値を超えると、そのRACノードはクラスタから削除されます。デフォルト値は200です。標準的なストレージ テイクオーバーやギブバックには、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ています。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）の変更が伝播されるまでの時間も長くする必要があります。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、大量のI/Oをキューに入れて再試行してから、投票ディスクI/Oを処理する必要があります。

ストレージを実際にテイクオーバーしたりギブバックしたりできない場合は、データベース サーバのケーブルを外すテストで影響をシミュレーションできます。

以下にNetAppの推奨事項を記載します。

- `disktimeout` パラメータはデフォルト値の200のままにします。
- RAC構成は必ず徹底的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。デフォルト値は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、NFSブートOS、およびVMDKファイルなどの仮想データストア上のブートドライブを搭載したホストが含まれます。

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 `misscount` を再開するのに必要な時間がしきい値を超えることがあります。結果、ブートLUNやGridバイナリへの接続がリストアされたあと、ノードは即座に削除されます。通常、削除とその後のリブートでは、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。

ローカル以外のブートドライブがある場合や、Gridバイナリをホストしているファイルシステムが `misscount` ある場合は、を同じ値に変更する必要が `disktimeout`あります。このパラメータを変更した場合は、ノードのフェイルオーバーの時間など、RACの動作への影響がないかどうかをさらにテストしてください。

以下にNetAppの推奨事項を記載します。

- `misscount` 次のいずれかの条件に該当する場合を除き、パラメータはデフォルト値の30のままにします。
  - 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ブートの場合
- このような場合は、ネットワークの中断がOSや `GRID_HOME` ファイルシステムへのアクセスに与える影響を評価します。このような中断によって原因Oracle RACデーモンが停止することがあり、 `misscount` ベースのタイムアウトや削除につながる可能性があります。タイムアウトのデフォルトは27秒です。これは `misscount` からを引いた値です `reboottime`。このような場合は、 `misscount` 200に増やして一致させ `disktimeout`ます。

Solarisクラスタ

アクティブ / パッシブ クラスタリング テクノロジであるSolaris Clusterは、他のクラスタウェアよりもはるかに高度に統合されています。このテクノロジは、ほぼプラグアンドプレイ機能を備えており、データベースやアプリケーションをクラスタ リソースとして簡単に導入できます。また、このテクノロジを使用すると、関連するIPアドレス、構成ファイル、ストレージ リソースなど、クラスタ内を簡単に移動できます。この緊密な統合の結果、Oracleはすべてのコンポーネントが適切に連携するように、Solarisクラスタ用の厳格な手順認定を取得しています。

ONTAPは、SAN環境のSolaris Clusterを幅広くサポートしています。詳細については、Interoperability Matrix Tool（IMT）を参照してください。

NFS環境でのサポートには制限があります。一般にNFSにサポート上の障壁はありませんが（自動マウントされたNFSホームディレクトリの使用など）、データベースをSolarisクラスタの制御下に置くことはできません。以前はNFSエージェントを使用できましたが、この製品は2012年10月にサポートが終了しました。Solarisクラスタのネイティブ機能を使用して、クラスタ化可能なカスタムサービスを構築することは可能ですが、ほとんどの環境では実現できません。ストレージなどのリソースを管理するスクリプトの作成に時間と手間がかかることがその理由です。

Veritas Cluster Server

Veritas Cluster Server（VCS）はSolaris Clusterと似ており、データベースまたはアプリケーションを導入サービスとしてパッケージ化し、アクティブ/パッシブ方式でクラスタに導入できます。

VCSとSAN

ONTAPは、SAN環境のVCSクラスタリングを幅広くサポートしています。詳細については、Interoperability Matrix Tool（IMT）を参照してください。

VCSとNFS

以前NetAppでは、クォーラム、監視、管理、フェンシング、NFSロック解除機能用にNFSクライアントを提供していましたが、主にこれらの機能が必要とされなくなったという理由から、NFSクライアントのサポートは2012年10月に終了しました。現在はVCSを使用して、標準でNFSファイルシステムを管理、監視できます。NAS環境のクォーラムの管理には、エージェントを必要としない複数の選択肢があります。

VCSとNFSフェンシング

アクティブ / パッシブ クラスタリングでは、フェンシングが1つの考慮事項です。フェンシングとは、ストレージ リソースを使用できるノードがクラスタの1つのノードに限定されることです。SAN環境の場合、SCSI永続的予約を使用してフェンシングを実装します。SCSI永続的予約では、ノードがLUNを独占的に制御することを要求できます。NFSの場合は、ファイルシステムのエクスポート オプションを変更して、複数のノードのリソースにアクセスできないようにします。2つの違いは、SAN環境では、ストレージ リソースを要求しているクラスタ内のノードによってフェンシングが実行されるのに対し、NAS環境では、ストレージ システム上でフェンシングを実行しなければならない点です。

NFSの場合、フェンシングは必須ではありません。SAN環境の場合、SANファイルシステムを複数のサーバにマウントするという単純な処理によってデータが即座に破損してしまうことが珍しくないため、フェンシングの実装はより重要です。NFSはクラスタ化されたファイルシステムであるため、複数のサーバに1つのファイルシステムを問題なくマウントできます。

多くのお客様は、VCSや、Hewlett Packard Enterprise Company（HPE）ServiceGuardやIBM PowerHAなどの同様の製品でアクティブ/パッシブクラスタリングを使用しています。フェンシングは設定されていません。1つのリソースが1つのノードだけで動作することが、クラスタ ソフトウェアによって保証されるためです。フェンシングが必要な場合は、スクリプト作成に少々手間をかけるだけで、クラスタ リソースの一部として導入できます。

サービスが開始されると、ストレージシステムに対してコマンドを実行して（a）ターゲットファイルシステムからすべてのノードへのアクセスを一時停止し、（b）サービスが開始される1つのノードへのアクセスを許可します。したがって、ターゲットファイルシステムでI/Oを実行できるのは1つのノードだけです。サービスのシャット ダウン時には、アクセスを削除するコマンドをストレージ システムに対して実行します。他のバリエーションも存在しますが、このプロセスは最も包括的なアプローチです。

これらのシステムのサポートは、NetAppプロフェッショナル サービスで提供しています。詳細は、NetAppの担当者にお問い合わせください。

VCSとNFSのロック解除

Oracle環境のNFSロックは、フェンシングの一種です。ターゲット ファイルにNFSロックを検出した場合、Oracleデータベースは起動しません。一般に、VCS環境でNFSロックを使用すると、VCSクラスタの正常な動作が妨げられます。ロックの解除が必要になるのは、正常にシャット ダウンしなかったノードのサービスを、別のノードがテイクオーバーするときのみです。Oracleデータベースのクリーン シャットダウン時には、ロックは削除されます。ノードがクラッシュした場合、ロックは維持されるため、データベースを再起動するにはロックの解除が必要です。

ほとんどのお客様は、最初からロックが作成されないようにするNFSマウント オプションを指定することで、NFSロックを無効にすることを選択しています。ロックの防止が望ましくない場合は、スクリプトを使用してロックを解除できます。フェンシングの場合と同様に、ロック解除スクリプトのサポートはNetAppプロフェッショナルサービスから利用できます。場合によっては、Rapid Response Engineeringを通じて完全にサポートされたオプションを利用できる場合もあります。詳細は、NetAppの担当者にお問い合わせください。

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、複雑なプロセスです。次の問題を理解している必要があります：

- 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を与える多数のOSおよびネットワーク設定

さらに、ストレージとは無関係のデータベース タスクを考慮する必要もあります。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なくなった場合、ストレージ パフォーマンスを最適化してもある時点でそのメリットはなくなります。

現在、データベースをご利用のお客様のほとんどがオールフラッシュアレイを選択しています。たとえば、2ノードのAFF8080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 / 書き込みの比率が75対25の場合、2台のAFF8080ノードを使用して30万回以上のランダムデータベースIOPSを、レイテンシ1ミリ秒未満で達成できます。この速度は、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可能性はほとんどなくなるでしょう。
- ネットワーク帯域幅が、パフォーマンスの制限要因としてますます一般化しつつあります。たとえば、回転式ドライブソリューションは、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。レイテンシによる制限がオールフラッシュ アレイによって取り除かれると、多くの場合ネットワークが新たな障壁となります。この変化は、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完全に使用できない場合、この制限により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は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ドライブを搭載したアレイのパフォーマンスを比較することはできません。テストの結果は大抵の場合意味がありません。
- 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も、データベースがストレージI/Oの制限を受けないことを考えれば、ほとんどの場合意味がありません。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。データベースの稼働時間の99%がCPU処理に費やされているとすれば、この違いは実環境では重要ではありません。ワークロードがストレージアレイのすべての機能を使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予想される統合システムでは、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 アレイの多くは桁外れのIOPSを謳っていますが、このレベルのIOPSはわずかなレイテンシによって意味をなさなくなります。オールフラッシュ アレイの場合、一般的な目安は1ミリ秒です。テスト方法

としては、IOPSの最大値を測定するよりも、平均レイテンシが1ミリ秒を上回らない範囲でストレージアレイが維持できるIOPSを確認することをお勧めします。

Oracle自動ワークロードリポジトリとベンチマーク

Oracleパフォーマンス比較のゴールド スタンダードは、自動ワークロード リポジトリ（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` コマンドを実行して生成されたレポートは、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イベントを分割する詳細なヒストグラムが含まれているため、最も包括的で価値のあるレポートとなります。

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ことが理想的です。長時間実行されているワークロードがある場合は、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。ただし、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、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、それ以外の方法では使用しないでください。

注： 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。

- `awrrpt.sql` コマンドを実行してレポートを生成します。レポートの出力形式には、テキストまたはHTMLを使用できます。
- 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刻を特定します。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。
- 可能な場合は、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。この手順では、 `awrrpt.sql` 15分間隔ごとにレポートを作成するために、を追加で実行する必要もあります。
- 問題が短いクエリの場合は、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番目の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。データベース サーバは、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、それ以外の方法では使用しないでください。
- パフォーマンスの問題が特定の時点で報告されているものの、他の状況では報告されていない場合は、比較のために良好なパフォーマンスを示す追加のAWRデータを提供します。

calibrate_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のドキュメントにあるように、このプロシージャはストレージのI/O性能をキャリブレート（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、I/Oを実行してデータベース処理をキャリブレートし、ホストに対して実行されるI/Oのレベルを最適化して効率性を高めることです。`calibrate_io` 処理で実行されるI/Oのタイプは実際のデータベースユーザI/Oを表していないため、結果は予測できず、再現さえできないことがよくあります。

SLOB2

Silly Little Oracle Benchmark（SLOB2）は、データベースのパフォーマンス評価に好んで使われるツールとなりました。Kevin Closson によって開発され、ここで入手できます。インストールと設定に数分かかりますが、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パターンを生成できます。オールフラッシュ アレイをI/Oで満たすことが可能な、新しいテストの1つです。生成するI/Oのレベルを大きく下げれば、IOPSは小さ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ションするのにも役立ちます。

Swingbench

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を使用することは困難です。NetAppが知るかぎり、Swingbenchを使ったテストで、AFFアレイに大きな負荷をかけられるほどのI/Oが生成された例はありません。ごく一部のケースでは、Order Entry Transaction（OET；オーダーエントリトランザクション）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。OETは、データベースに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レイテンシの可能性を実現するように、ホストとネットワークが適切に設定されていることを確認してください。

HammerDB

HammerDBは、特にTPC-CとTPC-Hのベンチマークをシミュレーションするツールです。テストを適切に実行するのに十分な大規模データセットの構築には時間がかかりますが、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ります。

Orion

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との互換性を維持するためにメンテナンスされていません。OSやストレージ設定との互換性がないことからOracle 10やOracle 11に使用されることはほとんどありませんでした。

Oracleはこのツールを書き直し、現在はOracle 12cにデフォルトでインストールされています。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、Oracleで使用されている正確なコードパスやI/O動作は使用しません。たとえば、大半のOracle I/Oは同時に実行されますが、I/O処理はフォアグラウンドで実行されるため、これはI/Oが完了するまでデータベースが停止することを意味します。ランダムI/Oでストレージシステムを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。

とは言え、たとえば特定のホスト / 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 システムの健全性評価にはOrionを使用できます。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、構成変更による影響を評価したりすることができます。使用するパラメータには、IOPS、スループット、レイテンシを考慮し、実際のワークロードを忠実にレプリケートする必要があります。

IBM AIX

このセクションでは、IBM AIXオペレーティング システム固有の構成について解説します。

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は連続したアトミックな（不可分な）I/Oを実行するため、大量のオーバーヘッドが発生します。そのため、同時I/Oを使用し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NetAppでは、`cio` マウント オプションを使用して、ファイルシステムで強制的に同時I/Oを実行することを推奨していました。ただしこの処理には欠点があるため、現在は推奨していません。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I/Oを実行できるようになりました。

同時I/Oを有効にする最良の方法は、 `init.ora` パラメータ `filesystemio_options` をに設定することです `setall`。これにより、Oracleが特定のファイルを開いて同時I/Oに使用できるようになります。

を `cio` マウントオプションとして使用すると、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（ファイルの複製、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恐れがあります。さらに、 `cio` Oracle GoldenGateやSAP BR * Toolsなどの製品は、特定のバージョンのOracleでマウントオプションを使用すると互換性がありません。

以下にNetAppの推奨事項を記載します。

- `cio` ファイルシステムレベルではマウントオプションを使用しないでください。代わりに、を使用して同時I/Oを有効にします `filesystemio_options=setall`。
- `cio` マウントオプションは、を設定できない場合にのみ使用して `filesystemio_options=setall` ください。

AIX NFSのマウントオプション

表1 と表2 に、AIX NFSのマウントオプションを示します。

表1）AIX NFSのマウントオプション–シングルインスタンス

ファイル タイプ	マウント オプション
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144
`ORACLE_HOM E`	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,intr

表2）AIX NFSのマウントオプション–RAC

ファイル タイプ	マウント オプション
`ADR_HOME`	rw,bg,hard,[[vers=3,vers=4,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,vers=4.1],proto=cp,timeo=600,rsize=262144,wsize=262144,noin tr, noac
CRS / Voting	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,noi ntr,noac
専用 `ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144

ファイル タイプ	マウント オプション
共有 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4,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 マウントオプションにが追加されたことです。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。

cio マウントオプションとinit.oraパラメータを使用した filesystemio_options=setall 場合も同様にホストのキャッシングが無効になりますが、は引き続き使用する必要があります。 noacNoac は、共有 ORACLE_HOME 環境でOracleパスワードファイルや spfile パラメータファイルなどのファイルの整合性を維持するために必要です。RACクラスタの各インスタンスに専用のORACLE_HOMEがある場合、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

表3 に、AIX JFS / JFS2のマウントオプションを示します。

表3）AIX JFS / JFS2のマウントオプション–シングルインスタンス

ファイル タイプ	マウント オプション
ADR_HOME	デフォルト
制御ファイル データファイル Redoログ	デフォルト
ORACLE_HOME	デフォルト

データベースを含むあらゆる環境でAIX hdiskデバイスを使用する場合は、パラメータqueue_depthをチェックします。このパラメータはHBAのキュー深度ではなく、個々のhdiskデバイスのSCSIキュー深度に関連するパラメータです。LUNの構成方法によっては、高いパフォーマンスを達成するにはqueue_depth の値が低すぎることがあります。テストでは、最適な値は64という結果が出ています。

HP-UX

このセクションでは、HP-UXオペレーティング システムに固有の構成について解説します。

HP-UX NFSのマウントオプション

表4 に、HP-UX NFSのマウントオプションを示します。

表4）HP-UX NFSのマウントオプション–シングルインスタンス

ファイル タイプ	マウント オプション
ADR_HOME	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,forcedirectio,nointr,suid
ORACLE_HOME	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid

表5）HP-UX NFSのマウントオプション–RAC

ファイル タイプ	マウント オプション
ADR_HOME	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,noac,suid
制御ファイル データファイル Redoログ	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac forcedirectio, suid
CRS / Voting	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid
共有 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4,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、RACにはマウント オプションにnoac とforcedirectio がある点です。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。init.ora パラメータを使用した filesystemio_options=setall 場合も同様にホストのキャッシングが無効になりますが、 noac とは引き続き使用する必要があります forcedirectio。

noac 共有 ORACLE_HOME 環境では、Oracleパスワードファイルやspfileなどのファイルの整合性を維持するために必要です。RACクラスタの各インスタンスに専用のORACLE_HOMEがある場合、このパラメータは必要ありません。

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、制御ファイルを格納し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
```

注: パラメータは 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を強制的に16以下にする場合は、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削除します。この手順はパフォーマンスの他の側面にも影響するため、の値を db_file_multiblock_read_count デフォルト値から変更する必要がある場合にのみ実行してください。

Linux

このセクションでは、Linux OSに固有の構成について解説します。

Linux NFS

スロットテーブル

LinuxでのNFSのパフォーマンスは、というパラメータに依存します `tcp_slot_table_entries`。このパラメータは、Linux OSで許容される未処理のNFS処理の数を制御します。

2.6の派生カーネル（RH5とOL5を含む）のほとんどではデフォルトは16ですが、このデフォルト値はパフォーマンス問題の原因となることがよくあります。反対の問題は、新しいカーネルでは `tcp_slot_table_entries` の値が上限に達していないため、過剰な要求でシステムがフラッディングすることでCan原因ストレージの問題が発生します。

解決策は、この値を静的に設定することです。NetApp NFSストレージとOracleデータベースを使用するLinux OSでは、値を128に設定してください。

RHEL 6.2以前でこの値を設定するには、次のエントリを入力し `in /etc/sysctl.conf`ます。

```
sunrpc.tcp_slot_table_entries = 128
```

また、2.6カーネルを使用するLinuxディストリビューションのほとんどにはバグがあり、起動プロセスは `/etc/sysctl.conf`、NFSクライアントがロードされる前にの内容を読み取ります。そのためNFSクライアントがロードされる時には、デフォルト値の16が使用されてしまいます。この問題を回避するには、を編集し `/etc/init.d/netfs` て `/sbin/sysctl -p` スクリプトの1行目でを呼び出し、`tcp_slot_table_entries` NFSがファイルシステムをマウントする前にが128に設定されるようにします。

RHEL 6.3以降でこの値を設定するには、RPC構成ファイルで次の変更を適用します。

```
echo "options sunrpc udp_slot_table_entries=64 tcp_slot_table_entries=128
tcp_max_slot_table_entries=128" >> /etc/modprobe.d/sunrpc.conf
```

Linux NFSのマウントオプション

表6 と表7 に、Linux NFSのマウントオプションを示します。

表6）Linux NFSのマウントオプション–シングルインスタンス

ファイル タイプ	マウント オプション
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr
`ORACLE_HOME`	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr

表7）Linux NFSのマウントオプション–RAC

ファイル タイプ	マウント オプション
`ADR_HOME`	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,actimeo=0
CRS / voting	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144

ファイル タイプ	マウント オプション
共有 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4,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 マウントオプションにが追加されたことです。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。init.ora パラメータを使用した filesystemio_options=setall 場合も同様にホストのキャッシングが無効になりますが、は引き続き使用する必要があります actimeo=0。

actimeo=0 共有 ORACLE_HOME 環境では、Oracleパスワードファイルやspfileなどのファイルの整合性を維持するために必要です。RACクラスタの各インスタンスに専用のORACLE_HOMEがある場合、このパラメータは必要ありません。

一般に、非データベース 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でマウントします。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 noac とは、actimeo=0 ファイルシステムレベルの先読みとバッファを無効にするため、できるだけ使用しないでください。これらのマウントオプションを使用すると、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もあります。

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あることを指摘しています。極端な例では、読み取りや書き込みなどの処理が、すべての処理の10%にまで低下することもあります。これは、 actimeo=0 noac Linux上でやを使用しているデータベースでは正常な動作です。これらのオプションを使用すると、Linux OSが原因を実行し、ファイルのメタデータがストレージシステムから定期的にリロードされます。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のキャッシュによって処理されます。読み取りや書き込みのように、ストレージ システムに本来の意味での要求を送信する純粋なIOPSとみなすことはできません。ただし、特にRAC環境では、このようなIOPSには負荷がかかります。この状況に対処するには、DNFSを有効にし、OSのバッファ キャッシュを迂回して不要なメタデータ処理を回避してください。

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台のサーバに複数回マウントされていて、（c）ネストされたNFSマウントが使用されている場合に必要です。このような構成は、主にSAPアプリケーションをホストしている環境に見られます。たとえば、NetAppシステム上の1つのボリュームのディレクトリを /vol/oracle/base とに配置できます /vol/oracle/home。/vol/oracle/base がマウントされ /oracle and /vol/oracle/home ている場合、がマウントされていると /oracle/home、同じソースに基づくNFSマウントがネストされます。

OSは /oracle 、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。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。こうすることで、OSキャッシングなどの処理は改善されますが、一方でDNFSの妨げとなります。DNFSがあるファイル（例えば /oracle/homeのspfile）にアクセスしなければならない場合、誤ってデータへの間違ったパスを使用する可能性があります。この場合、I/O処理は失敗します。このような構成では、 nosharecache ソース FlexVol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。このオプションを指定したファイルシステムには、Linux OSによって個別のデバイス ハンドルが割り当てられます。

Linux Direct NFSとOracle RAC

DNFSを使用すると、Linux OS上のOracle RACに特別なパフォーマンスメリットがあります。これは、RACでノード間の一貫性を確保するために必要なダイレクトI/Oを強制的に実行する方法がLinuxにないためです。回避策では actimeo=0 、マウントオプションを使用する必要があります。このオプションを使用すると、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。一方でこのオプションを指定すると、Linux NFSクライアントによって属性データが定期的に再読み取りされるため、レイテンシに影響が及

び、ストレージ コントローラの負荷が増大します。DNFSを有効にするとホストNFSクライアントが迂回されるため、この悪影響を回避できます。DNFSを有効にした場合、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 オプションを設定してLinuxにDNFSを使用する場合は、マルチパス サブネットを使用する必要があります。他のOSでは、 `LOCAL DONTROUTE` オプションとオプションを使用して1つのサブネットに複数のDNFSチャネルを設定することで、複数のDNFSチャネルを確立できます。Linuxではこの方法が適切に機能せず、予期しないパフォーマンスの問題が起きることがあります。Linuxでは、DNFSトラフィックに使用しているNICを、それぞれ別のサブネットに割り当ててください。

一般的なLinux SAN構成

圧縮のアライメント–パーティション

圧縮で最適な結果を得るには、8KBのドライブ境界にアライメントする必要があります。アライメントを確認するに `-u` は、オプションを指定してfdiskユーティリティを使用し、セクターに基づいてドライブを表示します。次の例を参照してください。

```
[root@jfs0 etc]# fdisk -l -u /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              36    20971519    10485742   83 Linux
Partition 1 does not start on physical sector boundary.
```

このパーティションは8KBでアライメントされていません。パーティションのオフセットは36セクターで、このオフセットは、優れたパフォーマンスを得るために必要な4KBの境界にアライメントされますが、8KBの境界にはアライメントされません。パーティションの開始位置を16セクターの倍数にして（512バイトx16=8,192）、パーティションをアライメントする必要があります。

以下に、正しくアライメントされたパーティションの例を示します。

```
[root@jfs0 etc]# fdisk -l -u /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              64    20971519    10485728   83 Linux
```

圧縮のアライメント–ファイルシステム

パーティションに加えて、ファイルシステムも8KBの境界にアライメントする必要があります。そのためには、ファイルシステムのブロック サイズが8KBである必要があります。Oracle ASMを使用している場合、ASMが実行するエクステントの割り当てとストライピングにより、8KBのアライメントが保証されます。

他のファイルシステムを使用する場合は、ブロック サイズを8KBに指定してください。ただしファイルシステムによっては指定できない場合もあります。

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のスケジューリングを低レベルで制御できます。Linuxのさまざまなディストリビューションでのデフォルト設定は大きく異なります。テストからは、通常はDeadlineを使用すると最適な結果が得られることがわかっていますが、NOOPの方が多少よい結果になることもあります。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。多くの構成ではCFQがデフォルトですが、これは、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を引き起こすことが実証されています。

I/Oスケジューラの設定方法については、関連する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、ネットワークが停止した際にシステムがクラッシュしてしまうことがありました。Linuxの最近のバージョンでは、OSとマルチパス デーモンのインストールの際に、OSがこの問題に対して脆弱なままに設定されることがあります。パッケージは正しくインストールされますが、リブート後に自動で起動するよう設定されません。

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@jfs0 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

これは、次のコマンドで修正できます。

```
[root@jfs0 iscsi]# chkconfig multipathd on
[root@jfs0 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。

NetApp Host Utilitiesのマニュアルに記載されているLinuxの設定には、I/Oのキューイングを無期限に行うマルチパス パラメータが含まれています。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されるまで、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、通常は望ましい方法です。

この無制限のキューイング動作により、ASMミラーリングに問題が発生します。ASMが代替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。

`multipath.conf` ASMミラーリングで使用するASM LUNのLinuxファイルで、次のパラメータを設定します。

```
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` 秒単位で計算されます。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秒後に発生し、I/Oエラーが発生しないようにする必要があります。この場合、障害グループはオフラインになります。

`no_path_retry` 値を小さくすると、ASMが代替障害グループに切り替えるのに必要な時間が短縮されますが、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慎重に監視することで、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は、再同期が比較的迅速に実行されると、ミラーを迅速に再同期できます。

詳細は、使用しているOracleソフトウェアのバージョンのASM Fast Mirror Resyncに関するOracleのマニュアルをご参照ください。

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するユーティリティです。ASMlibの第一の機能は、LUNやNFSベースのファイルに、ASMリソースであるという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は、Logical Blocks Per Physical Block Exponent（LBPPBE）というLUNのパラメータを検出します。最近までこの値はONTAP SCSIターゲットでは報告されていませんでしたが、現在は、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ます。この戻り値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デバイスが作成されるたびにASMヘッダーをスタンプします。

このプロセスでは、さまざまな方法でアップグレードや移行で原因の問題が発生する可能性があります。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。

例えば旧型のアレイでは、LBPPBEの値が0と報告されるか、まったく報告されません。ASMlibはこれを512バイトのブロック サイズと解釈します。新型のアレイは4KBのブロック サイズと解釈されると考えられますが、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。混在させると、2台のアレイの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することができなくなります。別のケースとしては、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 グループ間でファイルを複製しようとしても、許可されない可能性があります。

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以上で提供されています。このパッチを適用すると、`ORACLEASM_USE_LOGICAL_BLOCK_SIZE true /etc/sysconfig/oracleasm` 構成ファイルでパラメータをに設定できます。このオプションを設定すると、ASMlibはLBPPBEパラメータを使用できなくなるため、新しいアレイの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。

注： 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、ブロックサイズが512バイトのASMディスクグループを、ブロックサイズを4KBと報告す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、`ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 新しいLUNがASMlibでスタンプされる前にオプションを設定する必要があります。

によってすでにスタンプされているデバイスは `oracleasm`、新しいブロックサイズで再度スタンプされる前に再フォーマットする必要があります。まず、でデバイスの設定を解除し `oracleasm deletedisk`、次にでデバイスの最初の1GBをクリアし `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`ます。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` コマンドを使用して古いパーティションを削除するか、OSを再起動します。

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ことを確認する必要があります。ただし、データの移行は必要ありません。

ASMフィルタドライバのブロックサイズ

ASM Filter Driver（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は、ASMlibに似ています。ただし、Oracle以外のI/Oをブロックして、データを破損する可能性のあるユーザーエラーやアプリケーションエラーの可能性を減らすなどの追加機能も含まれています。

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。混在させると、2台のアレイの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することができなくなります。別のケースとしては、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 グループ間でファイルを複製しようとしても、許可されない可能性があります。

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。この設定は、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で論理ブロックサイズを強制的に使用するには、 `options oracleafd oracleafd_use_logical_block_size=1` /etc/modprobe.d//oracleafd.confファイルで設定します。

マルチパステンソウサイズ

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
        }
    }
```

注意： 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。

Linuxのxfs、ext3、ext4のマウントオプション

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ます。

Microsoft Windows

このセクションでは、Microsoft Windows OSに固有の構成について解説します。

NFS

Oracleは、Direct NFS ClientにMicrosoft Windowsをサポートしています。このことは、複数の環境のファイルの表示、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を管理する上でのメリットをもたらします。DNFSを使用したMicrosoft Windowsへのデータベースのインストールと設定については、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8、192バイト以上の割り当て単位が使用されていることを確認してください。一般的なデフォルトの4,096バイトの割り当て単位を使用すると、効率性が低下します。

Solaris

このセクションでは、Solaris OSに固有の構成について解説します。

Solaris NFSのマウントオプション

表8 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。

表8）Solaris NFSのマウントオプション–シングルインスタンス

ファイル タイプ	マウント オプション
ADR_HOME	rw,bg,hard,[vers=3,vers=4,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,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,llock,suid
ORACLE_HOME	rw,bg,hard,[vers=3,vers=4,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,suid

llock を使用すると、ストレージ システムでロックを取得およびリリースする際のレイテンシが取り除かれるため、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構成され、Oracleがそ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が必要です。これは非常に珍しい構成ですが、一部のお客様で使用されています。1つのインスタンスが誤って2回開始されると、Oracleは外部サーバのロック ファイルを検出できないため、データが破損します。NFSロックには、NFSバージョン3同様に保護機能はなく、単なる推奨事項です。

llock forcedirectio パラメータとパラメータを同時に使用することはできないため、filesystemio_options=setall init.ora が directio 使用されるようにファイルにを指定することが重要です。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。

表9 に、Solaris NFSのマウントオプションを示します。

表9）Solaris NFSのマウントオプション–RAC

ファイル タイプ	マウント オプション
ADR_HOME	rw、bg、hard、[vers=3、vers=4、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、vers=4.1]、proto=tcp、 timeo=600、rsize=262144、wsize=262144、nointr、noac、 フォースディレクトリ
CRS / Voting	rw、bg、hard、[vers=3、vers=4、vers=4.1]、proto=tcp、 timeo=600、rsize=262144、wsize=262144、nointr、noac、 フォースディレクトリ
専用 ORACLE_HOME	rw、bg、hard、[vers=3、vers=4、vers=4.1]、proto=tcp、 timeo=600、rsize=262144、wsize=262144、suid
共有 ORACLE_HOME	rw、bg、hard、[vers=3、vers=4、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、RACにはマウント オプションにnoac と forcedirectio がある点です。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。init.ora パラメータを使用した filesystemio_options=setall 場合も同様にホストのキャッシングが無効になりますが、 noac とは引き続き使用する必要があります forcedirectio。

actimeo=0 共有 ORACLE_HOME 環境では、Oracleパスワードファイルやspfileなどのファイルの整合性を維持するために必要です。RACクラスタの各インスタンスに専用のORACLE_HOMEがある場合、このパラメータは必要ありません。

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 オプションは、Snapshotバックアップの使用も維持します。

Solaris ZFS

パフォーマンスを最適化するために、Solaris ZFSをインストールして注意深く設定してください。

mvector

Solaris 11では、大規模なI/O処理の方法が変更され、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。この問題の詳細は、NetApp Bug Report 630173 「Solaris 11 ZFS Performance Regression.」をご参照ください。ソリューションは zfs_mvector_max_sizeというOSパラメータを変更することです。

rootとして次のコマンドを実行します。

```
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として実行すると簡単に元に戻すことができます。

```
echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

カーネル

信頼できるZFSパフォーマンスを実現するには、LUNのアライメント問題に対するパッチをSolarisカーネルに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に含まれています。ZFSではSolaris 10以降のみを使用してください。

LUNの設定

LUNを構成するには、次の手順を実行します。

1. タイプがsolarisのLUNを作成します。
2. IMTで指定されている適切なHost Utilitiesをインストールします。
3. Host Utilitiesの説明に記載されている手順に従ってください。このセクションでは基本的な手順について説明しますが、適切な手順については最新のドキュメントを参照してください。
   a. host_config ユーティリティを実行して sd.conf/sdd.conf ファイルを更新します。SCSIドライブがONTAPのLUNを正しく検出できるようになります。
   b. host_config ユーティリティの指示に従って、Microsoftマルチパス I/O（MPIO）を有効にします。
   c. リブートします。この手順は、変更をシステム全体に認識させるために必要です。
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。

zpool

zpoolを作成する前に、「LUNの設定」セクションの手順を実行してください。この手順を正しく完了しないと、I/Oのアライメントが原因で、深刻なパフォーマンス低下が起きる恐れがあります。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上に作成されたファイルシステムは、パラメータashiftで制御される実効ブロック サイズを使用します。これは、コマンドzdb -Cを実行することで確認できます。

の ashift 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。パフォーマンスを最適化するに ashift は、値が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zpoolのデータは、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、 ashift 次の手順に進む前にの値を確認してください。値が12以外の場合はLUNが正しく検出されていないので、zpoolを削除し、関連するHost Utilitiesのドキュメントに記載されているすべての手順が正しく実行され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の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が4Kデバイスとして正しく検出されても、LDOM上の仮想vdskデバイスにはI/Oドメインの設定が引き継がれません。そのLUNを基盤にしたvdskは、デフォルトの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。まず個々のLDOMにOracleのバグ15824910のパッチを適用し、設定オプションを追加できるようにします。このパッチは、現在使用されているすべてのSolarisバージョンに組み込み済みです。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 pool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";
```

この編集では、デバイスインスタンス1に4096のブロックサイズが割り当てられます。

別の例として、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` 次のように読み取ります。

```
"/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. 最終 `vdc.conf` ファイルには次の内容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。この手順は省略しないでください。ブロック サイズの変更が有効になるのはリブートのあとです。zpoolの設定を続行し、が `ashift` 前述のように正しく12に設定されていることを確認します。

ZIL

一般に、ZFS Intent Log（ZIL）を別のデバイスに配置する理由はありません。このログはメインのプールとスペースを共有できます。ZILを別に使用する主な理由は、最新のストレージ アレイで書き込みキャッシュ機能を備えていない物理ドライブを使用する場合です。

logbias

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

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

このパラメータを使用すると、全体的な書き込み量が削減されます。デフォルトでは、書き込みデータはまずZILにコミットされ、その後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。プレーンドライブ構成では、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が実行されます。

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、このアプローチは必要ありません。例えばレイテンシの影響を受けやすいランダム ライトが大量に発生するワークロードなど、ごくまれな状況においては、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。ログに書き込ま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るため、書き込み処理が二重に発生し、ライト アンプリフィケーションという状況を招きます。

ダイレクトI/O

Oracle製品を含め、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 キャッシュを迂回できます。ただしZFSファイルシステムでは想定する結果が得られず、ホストのバッファ キャッシュが迂回されてもZFS自体がデータをキャッシュし続けます。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。このような結果から、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になります。また、このような総合的なテストを使用してZFSのパフォーマンスを他のファイルシステムと比較することも非常に困難です。現実には、実際のユーザ ワークロードで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ベースのデータを、Snapshotをベースにバックアップ、リストア、クローニング、アーカイブする処理は、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としては、データファイルをに配置し zpool1 、アーカイブログ、制御ファイル、REDOログをに zpool2配置するのが最適です。このレイアウトでは、データベースがホット バックアップ モードに設定される標準のホットバックアップを実行し、続けてzpool1のSnapshotコピーを実行できます。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、ログがアーカイブされ、のSnapshotコピーが zpool2 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てzpoolを完全にオフラインにし、続けてSnapRestoreのリストア処理を実行します。その後、zpoolを再度オンラインにしてデータベースをリカバリできます。

filesystemio_options

Oracleパラメータは filesystemio_options ZFSでは動作が異なります。setall または directio を使用すると、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傍受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が想定よりも少なくなるため、パフォーマンス分析が困難になります。

まとめ

本ドキュメントの冒頭で述べたように、Oracle環境のストレージ構成は、環境による違いがあまりにも大きいため、本当の意味でのベストプラクティスはほとんどありません。データベースプロジェクトには、ミッションクリティカルなデータベースを1つ含めることも、5、000のレガシーデータベースや、数ギガバイトから数百テラバイトまでのサイズを含めることもできます。クラスタウェアや仮想化などのオプションを使用すると、さらに多様になります。

最適なソリューションは、環境の技術的な詳細と、プロジェクトの背景にあるビジネス要件の2つによって決まります。NetAppとパートナーのプロフェッショナル サービス エキスパートは、複雑なプロジェクトのサポートを提供しています。プロジェクト中はサポートが不要な場合も、NetAppが初めてのお客様には、プロフェッショナル サービスを利用して高度なアプローチの設計に役立てていただくことを強く推奨します。

付録A：古いNFS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ロックで問題が生じる場合があります。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、ロックの作成と解除に使用される2つの名前解決手法が微妙に異なることによって発生し、ネットワーク ロック マネージャ（NLM）とNFSクライアントという2つのプロセスが関係しています。NLMは uname -n ホスト名の決定にを使用し、 rpc.statd プロセスはを使用し gethostbyname() ます。OSが古いロックを適切に解除するためには、この2つのホスト名が一致していなければなりません。例えば、dbserver5によって所有されるロックを検索している場合に、dbserver5.mydomain.orgという名前のホストでロックが登録されていたとします。gethostbyname() がと同じ値を返さない場合 uname -a、ロック解除プロセスは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";
```

gethostbyname が一致しない場合 unameは、ロックが古い可能性があります。以下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、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 /etc/hostsます。例えば、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
```

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記述順序を次のように変更します。

```
10.156.110.201 dbserver5 dbserver5.mydomain.org loghost
```

gethostbyname() では dbserver5 、の出力に一致する短いホスト名が返される unameようになりました。このため、サーバがクラッシュするとロックは自動的に解除されます。

付録B：WAFLのアライメントの検証

優れたパフォーマンスを達成するには、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、だからと言ってすべての処理が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のI/Oがこの4KB単位にどう対応しているかを意味します。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。Oracleのブロックが2KBずれて配置されている場合、このブロックは、4KBブロックの半分と次の4KBブロック全体、さらに3つ目の4KBブロックに配置されます。この状態は、パフォーマンスの低下を招きます。

NASファイルシステムの場合、アライメントは問題となりません。OracleデータファイルはOracleブロックのサイズを基にファイルの開始位置にアライメントされるため、常に8KB、16KB、32KBのブロック サイズでアライメントされます。すべてのブロック処理は、ファイルの開始から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、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ネイティブの4KBセクターを使用する物理ドライブ向けに設計されているためです。

ただし、例外がいくつかあります。4KB I/O向け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可能性があります。または、パーティション作成時のユーザエラーにより、4KB単位でないオフセットが発生した可能性があります。

以下はLinux固有の例ですが、手順はどのOSにも該当します。

アライメント

以下に、パーティションが1つの単一のLUNでアライメントをチェックする例を示します。

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@jfs0 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@jfs0 iscsi]#
```

以下のコマンドを使用して、アライメントを正確にチェックできます。

```
[root@jfs0 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
```

出力からは、セクターが512バイト単位で、パーティションの開始位置がこのセクター32個分であることがわかります。この測定値の合計は32 x 512 = 16、834バイトです。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をいくつか実行します。`iflag` 引数を使用して、I/Oがバッファされずに同期されるようにすることが重要です。

注：このコマンドには注意してください。`if of` 引数と引数を逆にすると、データが破棄されます。

```
2)  [root@jfs0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000 bs=4096
3)  1000+0 records in
4)  1000+0 records out
5)  4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. 統計を終了し、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` バケット内にある必要があります。これは、I/Oが4KBのブロック境界にアライメントされていることを示します。

```
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

以下に、ミスアライメントI/Oの例を示します。

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。このミスアライメントは、最新のOSではデフォルトの動作ではありません。

```
[root@jfs0 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セクターのオフセットで作成されています。「Aligned」セクションで説明した手順を繰り返します。 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%
```

明らかにミスアライメントであることがわかります。I/Oの大部分が `.1` バケットに収まり、想定されるオフセットと一致します。パーティションの作成時に、最適なデフォルトから512バイトうしろに作成したので、ヒストグラムは512バイトでオフセットされています。

また、`read_partial_blocks` 統計値はゼロではありません。つまり、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 ログは、I/Oパターンが異なります。例えば、Redoロギングでは1つのファイルが繰り返し上書きされます。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はヒストグラムのすべての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とはなりません。ただし、Redoロギング率がきわめて高い場合は、4KBのブロック サイズを使用するとパフォーマンスが向上します。この場合は、RedoロギングLUNを正しくアライメントされていることを推奨しますが、優れたパフォーマンスにとっては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。

詳細情報の入手方法

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

- ONTAPおよびONTAP System Managerのドキュメントリソース
  https://www.netapp.com/data-management/oncommand-system-documentation/
- NetAppの製品ドキュメント
  https://www.netapp.com/support-and-training/documentation/

バージョン履歴

バージョン	日付	ドキュメント バージョン履歴
バージョン1.0	2020年3月	初版

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

機械翻訳に関する免責事項

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

著作権に関する情報

商標に関する情報

TR-3633-0521-JP