ストレージ構成

Enterprise applications

NetApp
May 09, 2024

目次

ストレージ構成

FC SAN

OracleデータベースI/OのLUNアライメント

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

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

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

Linux環境では、ドライブデバイス全体に論理ボリュームグループを構築します。パーティションが必要な場合は、次のコマンドを実行してアライメントを確認します。 `fdisk -u` 各パーティションの開始が8の倍数であることを確認します。つまり、パーティションは8の倍数の512バイトセクター（4KB）から開始されます。

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

ミスアライメントノケイコク

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

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

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

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

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

Oracleデータベースの**LUN**のサイジングと数

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

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

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

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

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

LUN数

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

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

- NetAppの推奨事項*：

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

Oracleデータベースの**LUN**配置

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

個のボリューム

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

る）を使用することです。

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

ボリュームと**LUN**

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

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

ボリューム、**LUN**、**Snapshot**

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

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

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

ボリューム、**LUN**、**SnapMirror**

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

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

ボリューム、**LUN**、**QoS**

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

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

マルチボリュームレイアウト

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

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

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

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

Oracleデータベースの**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容量の上限に達した場合は、次の2つの方法で使用可能なスペースを増やすことができます。

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

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

Oracleデータベースでの**LVM**ストライピング

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

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

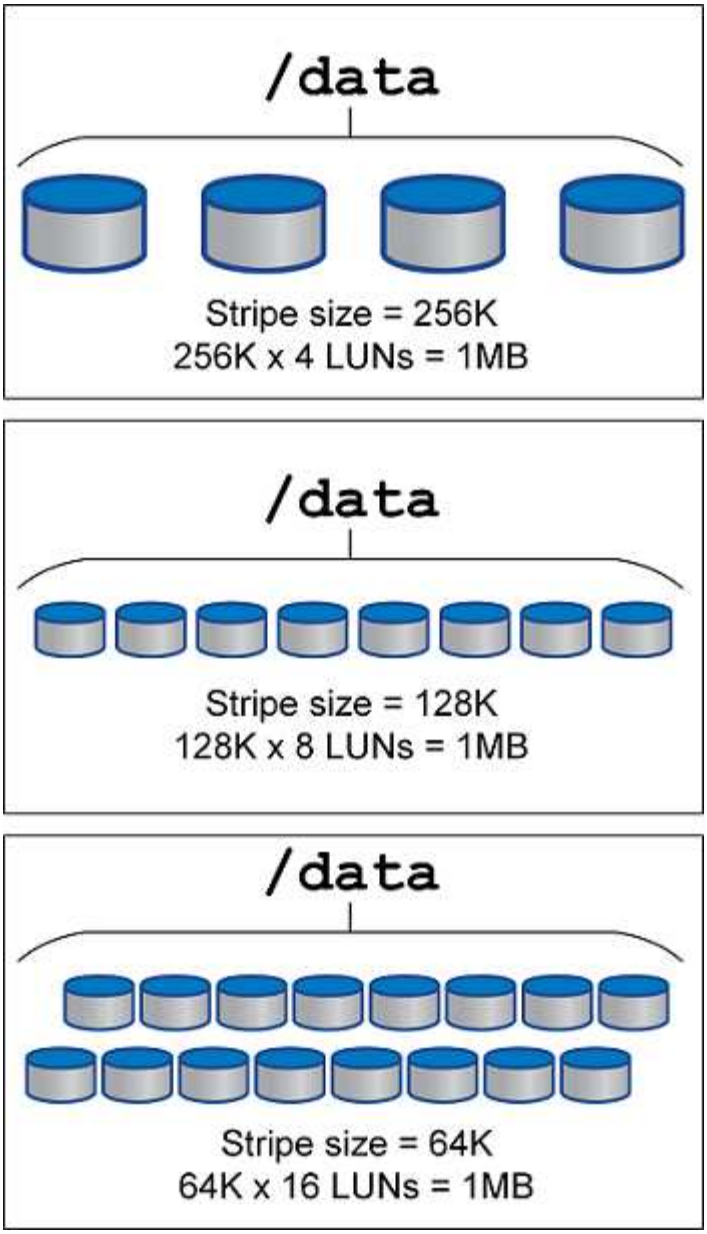
回転式ドライブを使用したストライピングは、I/Oパターンを事前に把握しておく必要があったため、より困難でした。ストライピングが実際のI/Oパターンに合わせて正しく調整されていない場合、ストライピングされた構成ではパフォーマンスが低下する可能性があります。Oracleデータベース、特にオールフラッシュ構成では、ストライピングは設定がはるかに簡単で、パフォーマンスが劇的に向上することが実証されています。

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

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

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

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



NFS

Oracleテタヘスヨウノ**NFS**ノセツテイ

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

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

NFSハアシヨン

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

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

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

Linux NFSv3 TCPスロットテーブル

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

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

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

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

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

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場合もあります。一方では、Linux OSが問題で処理できる以上のI/Oを試行すると、I/Oレイテンシが増加します。

ADRと**NFS**

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

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

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

nfs-rootonlyおよび**mount-rootonly**

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

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

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

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

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

パラメータは次のように変更できます。

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

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

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

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

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

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

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

NFSv4 / 4.1構成

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

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

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

NFSv4ドメイン

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

ONTAP SVMでNFSv4ドメイン名が設定されます。

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

ホストのNFSv4ドメイン名は、 `/etc/idmap.cfg`

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

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

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

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

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

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

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

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

Oracle directNFS

Oracleデータベースでは、NFSを2つの方法で使用できます。

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

2つ目の方法はOracle Direct NFS（dNFS）です。これは、Oracleデータベースソフトウェアに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に正しい設定があるかぎり、DBAチームやエンドユーザはdNFSを透過的に使用できなければなりません。

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

Direct NFS

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

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

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

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

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

Direct NFSとホストファイルシステムへのアクセス

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

シングルインスタンスデータベースのマウントオプションを使用すると、ファイルおよびディレクトリの属性のキャッシュが有効になり、ディレクトリの内容もキャッシュされます。そのため、dNFSでファイルが作成される可能性があり、OSがディレクトリの内容を再読み取りしてファイルがユーザに表示されるまでに少し

時間がかかります。これは通常は問題になりませんが、まれに、SAP BR * Toolsなどのユーティリティで問題が発生することがあります。この場合は、マウントオプションをOracle RACの推奨事項に変更して、問題に対処してください。この変更により、ホストのキャッシュがすべて無効になります。

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

次の注を参照： `nosharecache` インチ "Linux NFSのマウントオプション" 通常とは異なる結果を生成する可能性があるLinux固有のdNFS問題の場合。

Oracleデータベースと**NFS**のリースとロック

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

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

NFSv4のステートフル

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

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

NFSv4ロック

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

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

NFSv4リース

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

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

定します。

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

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

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

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

```
 *Question:* What happens if you lose network connectivity for 31 seconds?
```

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

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

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

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

これが発生した場合のOracleアラート・ログの出力例を次に示します

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

syslogを確認すると、次のエラーのいくつかが表示されます。

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

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

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

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

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

NFSv4猶予期間

NFSv3サーバをリブートすると、ほぼ瞬時にIOを処理できるようになります。それはクライアントの状態を維持することではありませんでした。そのため、ONTAPのテイクオーバー処理はほぼ瞬時に実行されることがよくあります。コントローラがデータの提供を開始する準備ができた時点で、ネットワークにARPを送信し、トポロジの変更を通知します。通常、クライアントはこれをほぼ瞬時に検出し、データの流れを再開します。

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

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

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

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

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

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

リースタイムアウトと猶予期間

猶予期間とリース期間が接続されます。前述したように、デフォルトのリースタイムアウトは30秒です。つまり、NFSv4クライアントは少なくとも30秒ごとにサーバにチェックインする必要があります。そうしないと、リースとロックが失われます。この猶予期間はNFSサーバがリース/ロックデータを再構築できるようにするためのもので、デフォルトは45秒です。ONTAPでは、猶予期間をリース期間より15秒長くする必要があります。これにより、リースを30秒以上更新するように設計されたNFSクライアント環境では、再起動後にサーバにチェックインできます。猶予期間を45秒に設定することで、少なくとも30秒ごとにリースを更新することを期待するすべてのクライアントが確実に更新する機会を得ることができます。

30秒のタイムアウトが許容されない場合は、リース期間を延長することもできます。60秒のネットワーク停止に耐えるためにリースタイムアウトを60秒に延長する場合は、猶予期間を少なくとも75秒に延長する必要があります。ONTAPでは、リース期間より15秒長くする必要があります。つまり、コントローラフェイルオーバー中にIOが一時停止する時間が長くなります。

これは通常は問題ではありません。一般的なユーザはONTAPコントローラを年に1~2回更新するだけで、ハードウェア障害による計画外フェイルオーバーは非常にまれです。また、ネットワークに60秒のネットワーク停止が発生する可能性があり、リースタイムアウトを60秒にする必要がある場合は、まれにストレージシステムのフェイルオーバーに異議を唱えず、75秒の一時停止も発生する可能性があります。ネットワークが60秒以上頻繁に一時停止していることをすでに認識しています。

Oracleデータベースでの**NFS**キャッシュ

次のマウントオプションが存在すると、ホストのキャッシュが無効になります。

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

これらの設定は、ソフトウェアのインストール、パッチ適用、およびバックアップ/リストアの処理速度に重

大な悪影響を及ぼす可能性があります。場合によっては、特にクラスタ化されたアプリケーションでは、クラスタ内のすべてのノードにキャッシュの一貫性を提供する必要があるため、これらのオプションが必然的に必要になることがあります。それ以外の場合、顧客はこれらのパラメータを誤って使用し、結果は不要な性能の損傷です。

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

Oracleデータベースでの**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。

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

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

特定のONTAP SVMの最大転送サイズは、次のように変更できます。

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

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。

Oracleデータベースと**NVFAIL**

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

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

スク上のデータの状態を反映しなくなります。この不整合により、データが破損します。

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

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

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

ASM再生ユーティリティとONTAP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、DBAはデータが削除されたあとにストレージアレイのスペースを再生できます。ONTAPはゼロをインターセプトし、LUNからスペースの割り当てを解除します。ストレージシステム内にデータが書き込まれていないため、再生プロセスは非常に高速です。

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

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

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

著作権に関する情報

商標に関する情報