テクニカル レポート

NFS in NetApp ONTAP
ベストプラクティスおよび実装ガイド

NetApp
Justin Parisi / Elliott Ecton
2023年6月| TR-4067

概要

本ドキュメントでは、NetApp® ONTAP®におけるNFSの基本概念、サポート情報、設定のヒント、およびベストプラクティスについて説明します。このガイドでは、通貨と長さに応じて利用可能な最新のONTAPバージョンについて説明します。ほとんどの場合、このガイドに記載されているコマンドとベストプラクティスはすべてのONTAPバージョンに適用されます。

旧バージョンのONTAPでは結果が異なる場合があります。必要に応じて、ご使用のONTAPバージョンの製品ドキュメントを参照してください。

目次

表一覧

図一覧

NetApp ONTAPでのNFSの基本概念

対象読者および前提条件

本テクニカル レポートは、ストレージ管理者、システム管理者、およびデータセンター管理者を対象としており、以下の点に精通していることを前提としています。

- NetApp ONTAPデータ管理ソフトウェア
- ネットワーク ファイル共有プロトコル（特にNFS）

このドキュメントには、アドバンスレベルおよび診断レベルのコマンドがいくつか含まれています。これらのコマンドを使用する際には、細心の注意を払ってください。コマンドの使用に関する質問や懸念事項については、NetAppのサポートにお問い合わせください。

NFSのサポートを表明

次のセクションでは、NFSバージョンのサポートとONTAPでのクライアントのサポートについて説明します。

NFSバージョンのサポート

表1 に、ONTAPでサポートされるNFSバージョンと、プロトコルでサポートされるONTAPリリースを示します。

表1）ONTAPでサポートされるNFSバージョン

NFSバージョン	サポートされるONTAPバージョン
NFSv2	ONTAP 8.2以前
NFSv3	すべてのONTAPリリースでサポート
NFSv4.0	ONTAP 8.1以降
NFSv4.1（pNFS使用）	ONTAP 8.1以降
NFSv4.2	ONTAP 9.8（基本プロトコルのサポート） ONTAP 9.9.1（NFSラベル付き）

NFSクライアントのサポート

NFSv3クライアントはInteroperability Matrix（IMT）には表示されません。ONTAPは、Internet Engineering Task Force（IETF；インターネット技術特別調査委員会）が承認したRequest for Comments（RFC）ドキュメントで定義されている標準に準拠したすべてのNFSクライアントをサポートしています。NFSv4.xのクライアントのサポートは、各ベンダーに記載されているクライアントのサポートにも依存します。IMTではNFSv4.1クライアントがサポートされています。

NetApp IMTはこちらからご覧いただけます。

ONTAPでサポートされている各NFSバージョンの最新のRFCを次に示します。

- RFC-1813：NFSv3
- RFC-7530：NFSv4.0
- RFC-5661：NFSv4.1
- RFC-7862：NFS4.2

Windows NFSのサポート

ONTAP 8.2.3および8.3.1以降では、Windowsクライアントを使用するNFSのサポートが追加されています。Windows NFSはRFC規格に準拠していないため、このプロトコルを使用するためには追加の考慮事項が必要です。詳細については、「WindowsでのNFS」を参照してください。

NFSセキュリティのサポート

表2 に、保存中と転送中の両方のNFSを保護するためにサポートされているセキュリティ方式を示します。ONTAPの特定のセキュリティ機能の詳細については、TR-4569：『Security Hardening Guide for NetApp ONTAP』を参照してください。

- **保存データセキュリティ** とは、インプレースにあるデータのセキュリティのことです。
- **転送中セキュリティ** とは、ネットワーク経由で転送されるデータのセキュリティのことです。

表2）NFSセキュリティのサポートの詳細

NFSデサポートサレテイルホゾンセキユリテイ	NFSデサポートサレル転送中セキュリティ
• NetApp Volume Encryption（NVE） • NetApp Aggregate Encryption（NAE） • 自己暗号化ドライブ（SED） • NetAppストレージ暗号化ドライブ（NSE） • エクスポートポリシーとルール • Access Control List（ACL;アクセス制御リスト） • アイデンティティ管理（ユーザ、グループ、ファイル所有権）	• Kerberos（krb5、krb5i、krb5p） – サポートされる暗号化タイプは、AES-256、AES-128、3DES、DESです。

注: ONTAPでは、SSH経由のNFSとstunnel経由のNFSはサポートされていません。

NFSの機能

各NFSバージョンでは、プロトコルに新しい機能が追加され、運用、パフォーマンス、ビジネスのユースケースが強化されます。次の表に、ONTAPでサポートされているNFSの機能と、その機能に関連付けられているNFSのバージョン情報を示します。

表3）NFS機能のサポート

NFSバージョン	利用可能な機能
全バージョン	• ボリュームレベルおよびqtreeレベルのエクスポートルール • NFS処理の最大補助GID数：1、024 • 64ビットノファイルID • ネームスペース全体でボリュームをジャンクションして疑似ファイルシステムを作成する機能 • UNIX形式のモードビット権限（rwx） • TCPおよびUDP（NFSv4.xはTCPのみ） • ポートの変更 • ポート範囲の制限：1~1024（mount-rootonly、nfs-rootonly） • ファイルシステム間でのFSIDの変更 • showmount（showmount詳細および制限事項についてはを参照） • NFSクライアントとボリュームのマッピング • マルチプロトコルのNASアクセス（CIFS / SMB、NFS） • UNIX IDマッピング用のLightweight Directory Access Protocol（LDAP）/ Network Information Service（NIS） • ネットグループ • Windows NFS
NFSv3	• NFSv3のすべてのRFC標準機能をサポート
NFSv4.0 / 4.1	• ACL（最大1、024） • 委譲 • イコウ

NFSバージョン	利用可能な機能
	• リファーラル • リースタイムアウトの設定 • pNFS（v4.1）
NFSv4.2	• ONTAP 9.8での基本的なプロトコルのサポート • ラベル付きNFS（ONTAP 9.9.1、ゲストモードおよび限定サーバモードのみ）

次の表に、NFSで現在サポートされていない機能を示します。

表4）サポートされないNFS機能

NFSバージョン	サポートされない機能
全バージョン	• POSIX ACL • サブディレクトリのエクスポート • SSSD動的UID
NFSv3	• 拡張属性（RFC仕様ではない）
NFSv4.0 / 4.1	• セッショントランキング/マルチパス
NFSv4.2	• ライブファイル移行（Flexファイル） • スパースファイル • スペース リザベーション • IO_advise • アプリケーションデータホール • サーバ側のコピー • ラベル付きNFSのフルモード

NFSサーバオプション

ONTAPのNFSサーバは、サーバオプションを使用して設定できます。デフォルトでは、ベストプラクティスを念頭に置いて設定されているため、特別な変更は必要ありません。ただし、場合によっては（NFSv4.xサポートの有効化など）、NFSオプションの変更が必要になります。

これらのデフォルト値、および使用可能なすべてのNFSオプションと説明については、使用しているバージョンのONTAPのマニュアルページ、または `man nfs modify` CLIからを実行して説明しています。

rootonlyオプション-nfsrootonlyおよびmountrootonly

`rootonly` 信頼されないクライアントアクセスを回避するためのオプションが追加されました。信頼されていないクライアント（エクスポートルールに含まれていないクライアント）は、信頼されたクライアントへのSSHトンネリングを使用してデータにアクセスする可能性があります。ただし、この要求は信頼できないポート（1,024より大きいポート）から送信されます。これは、アクセスする意図のないクライアントにとってバック ドアとなります。

そのため、 `rootonly` オプションの有効化と無効化は必要に応じて異なります。つまり、環境でNFSが正常に機能するためにより多くのポートが必要か、それとも信頼できないクライアントによるマウントへのアクセスを防止する方が重要かということです。

NFSv4.xやKerberos認証を利用してNFSエクスポートへのアクセスのセキュリティを強化することも1つの妥協案です。TR-4616：『NFS Kerberos in ONTAP』では、NFS Kerberosの使用方法について説明しています。

このようなシナリオでは `mount-rootonly` 、 `nfs-rootonly` オプションやオプションを使用することで問題を軽減できます。

クライアントのポート使用状況を確認するには、次のコマンドを実行します。

```
# netstat -na | grep [IP address]
```

クラスタでのポートの使用状況を確認するには、次のコマンドを実行します。

```
cluster::> network connections active show -node [nodename] -vserver [vservername] -service nfs*
```

これらのオプションの詳細と、多数のNFSクライアントがある環境でこれらのオプションが役立つ状況の例については、「多数のNFSクライアントでのネットワークポートの枯渇」を参照してください。

showmount

8.3より前のONTAPでは、`showmount` NFSクライアントからのコマンドでエクスポートパスを公開できません。この制限は、パフォーマンスを考慮して設計されたものです。ONTAPクラスタには数千ものエクスポートルールが設定される可能性があるため、すべてのエクスポートに対するクエリの処理負荷が高くなる可能性があります。また、エクスポートはフラット ファイル形式ではなく、ルールとしてボリュームに適用されるため、エクスポート パスとエクスポート ルールは別々の場所に存在します。

ONTAPでshowmountが無効になっているクライアントのshowmount-eの例：

```
[root@nfsclient /]# showmount -e x.x.x.a
Export list for x.x.x.a:
/ (everyone)
```

SVMのvsrootボリュームはにマウントされて `/`います。このボリュームは、showmountクエリで返されます。他のすべてのボリュームはそのマウントポイントの下にマウントされ、showmount NFSオプションを無効にしてもクライアントには返されません。

showmountクエリの実行時の動作

showmountは、NFSv3のMOUNTプロトコルを利用して、NFSサーバへのエクスポートクエリを問題します。マウント ポイントがリスンしていないか、ファイアウォールによってブロックされている場合、あるいはNFSサーバでNFSv3が無効になっている場合は、showmountクエリが失敗します。

```
# showmount -e x.x.x.a
mount clntudp_create: RPC: Program not registered
```

次の例は、 `showmount` オプションがdisabledに設定されたONTAP内のデータLIFに対して実行されたコマンドのパケットトレースからの出力を示しています。

```
x.x.x.x x.x.x.a MOUNT  170    V3 EXPORT Call (Reply In 17)
Mount Service
Program Version: 3
V3 Procedure: EXPORT (5)

x.x.x.a x.x.x.x MOUNT  202    V3 EXPORT Reply (Call In 16)
Mount Service
Export List Entry: /unix ->
```

注: トレースは、サーバーが戻ったことを示してい `/unix ->`ます。ただし、このエクスポート パスはルール セットに特定のクライアントが指定されています。

```
cluster::> vol show -vserver NFS83 -junction-path /unix -fields policy
  (volume show)
vserver volume policy
---------- -------- -----------
NFS83   unix   restrict

cluster::> export-policy rule show -vserver NFS83 -policyname restrict
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
NFS83        restrict        1       any      x.x.x.y         any
```

showmount機能でクライアントの一致が必要な場合は、Toolchestのshowmountユーティリティがその機能を提供します。

ONTAP 8.3以降のshowmount

showmount機能は、Oracle OVMなどの一部のアプリケーションが正常に動作するために必要なため、これらのアプリケーションを適切にサポートするためにONTAP 8.3以降でshowmountのサポートが追加されました。

この機能はデフォルトでは無効になっています。次のコマンドで有効にすることができます。

```
cluster::> nfs server modify -vserver NFS -showmount
    enabled disabled
```

注： ONTAPでshowmountを使用するには、親ボリューム（vsrootまたは/を含む）がshowmountを実行するクライアント/ユーザに読み取りアクセスまたはトラバースアクセスを許可する必要があり、vsroot（/）はUNIXセキュリティ形式を使用する必要があります。

有効にすると、クライアントがデータLIFにエクスポート パスを照会できるようになります。ただし、clientmatch（クライアント、ネットグループなどからのアクセス）の情報は出力されず、エクスポートポリシールールセットでクライアントが指定されている場合でも、各パスにはアクセス権があるすべてのユーザが反映されます。

clustered ONTAP 8.3以降でのshowmountの出力例：

```
# showmount -e x.x.x.a
Export list for x.x.x.a:
/unix       (everyone)
/unix/unix1 (everyone)
/unix/unix2 (everyone)
/           (everyone)
```

注： Windows NFSを使用している場合は、ファイルおよびフォルダの名前変更に関する問題を回避するためにshowmountを有効にする必要があります。

showmountのキャッシュ

クライアントから実行されたshowmountは、クラスタ上のNFSサーバに対して情報を要求します。エクスポートリストはサイズが大きくなる可能性があるため、クラスタはこの情報のキャッシュを保持して、NFSサーバに対する要求数を削減します。

volume unmount コマンドまたはONTAP System Managerを使用してクラスタネームスペース（「クラスタネームスペース」を参照）からボリュームをアンマウントしてもキャッシュは更新されないため、エクスポートされたパスは期限切れになるかフラッシュされるまでキャッシュに残ります。

showmountキャッシュをフラッシュします。

```
cluster::> export-policy cache flush -vserver SVM -cache showmount
```

キャッシュがフラッシュされるのは、ログインしているノードだけです。たとえば、node1の管理LIFにログインしている場合は、node1のキャッシュがフラッシュされます。つまり、キャッシュフラッシュの恩恵を受けるのは、node1のローカルなデータLIFに接続しているクライアントだけです。他のノード上のキャッシュをフラッシュするには、当該ノードのノード管理LIFにログインします。コマンドを実行すると、フラッシュ中のノードが表示されます。

```
cluster::> export-policy cache flush -vserver SVM -cache showmount

Warning: You are about to flush the "showmount" cache for Vserver "SVM" on node "node1", which
will result in increased traffic to the name servers. Do you want to proceed with flushing the
         cache? {y|n}: y
```

ネームスペースの概念

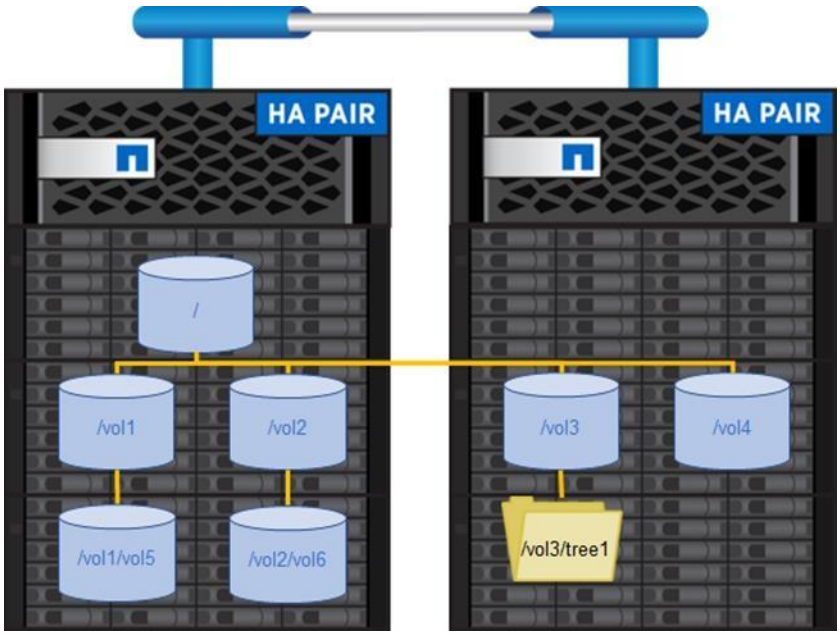
このセクションでは、NFS環境における「ネームスペース」の概念と、ONTAPがNFSクライアントにグローバルネームスペースを提供するための複数のソリューションを提供する仕組みについて説明します。

クラスタネームスペース

ONTAPのネームスペースは、拡張性に優れたパフォーマンスと容量を提供するために、クラスタ内の複数のノードでホストされるファイルシステムの集まりです。各SVMには、単一のルート ボリュームで構成されるファイル ネームスペースが1つあります。このネームスペースはの場所から始まります。後続のボリュームおよびqtreeはすべてがをトラバースし、ボリュームオプションで定義されたエクスポートパスを持ちます - `junction-path`。SVMネームスペースは1つ以上のボリュームで構成できます。ボリュームはジャンクションでリンクされ、あるボリュームの名前付きジャンクションinodeから別のボリュームのルートディレクトリに接続されます。クラスタには複数のSVMを含めることができますが、各SVMに割り当てられるvsrootとが1つだけであるため、各SVMには一意のファイルシステムIDのセットが割り当てられます。
これにより、複数のSVMにあるボリュームでファイルシステムID /ファイルハンドルが共有されるのを防ぎ、マルチテナント環境でのNFSエクスポートのマウントに関する問題を回避できます。

SVMに属するすべてのボリュームは、エクスポートパスを使用して、そのクラスタのグローバルネームスペースにリンクされます。クラスタ ネームスペースは、クラスタ内の単一ポイントでマウントされます。クラスタ内のクラスタネームスペースの最上位ディレクトリ（「/」）は統合されたディレクトリで、クラスタ内の各SVMネームスペースのルートディレクトリのエントリが含まれています。ネームスペースには、NetApp FlexVol® ボリュームまたはNetApp ONTAP FlexGroupボリュームを使用できます。

図1）クラスタネームスペース



ネームスペースの保護

vsrootボリュームは、複数のノードからSVMにアクセスできても、クラスタ内の単一のノードにのみ存在します。vsrootはNFSクライアントがネームスペースをトラバースする方法であるため、NFSの処理にとって非常に重要です。

```
cluster::> vol offline -vserver NFS -volume vsroot

Warning: Offlining root volume vsroot of Vserver NFS will make all volumes on that Vserver
inaccessible.
Do you want to continue? {y|n}: y
Volume "NFS:vsroot" is now offline.
```

vsrootボリュームが何らかの形で使用できない場合、vsrootボリュームがファイルシステムをトラバースする必要があるときにNFSクライアントに問題が発生します。

これには、次の動作が含まれます（ただし、これらに限定されません）。

- マウント要求がハングします。
- /がマウントされている場合は、別のボリュームとの間でトラバーサルを行い、lsとsonを実行してハングアップします。
- ボリュームがオンラインに戻っても、マウントがビジー状態であるためにアンマウント処理が失敗することがあります。
- ボリュームがすでにでマウントされている（/vol1など）場合でも、読み取り/書き込み/表示は成功します。

ONTAPの負荷共有ミラーを使用すると、ONTAPのSnapMirror機能を活用してvsrootの耐障害性を高めることができます。

注： 負荷共有ミラーはvsrootボリュームでのみサポートされます。データボリューム間で負荷を共有するには、代わりにNetApp FlexCacheボリュームを使用することを検討してください。

vsrootボリュームで負荷共有ミラーを使用できる場合、NFSv3処理は負荷共有ミラーデスティネーションボリュームを利用してファイルシステムをトラバースできます。負荷共有ミラーを使用している場合は `.admin` 、NFSマウント内のフォルダからソースボリュームにアクセスできます。

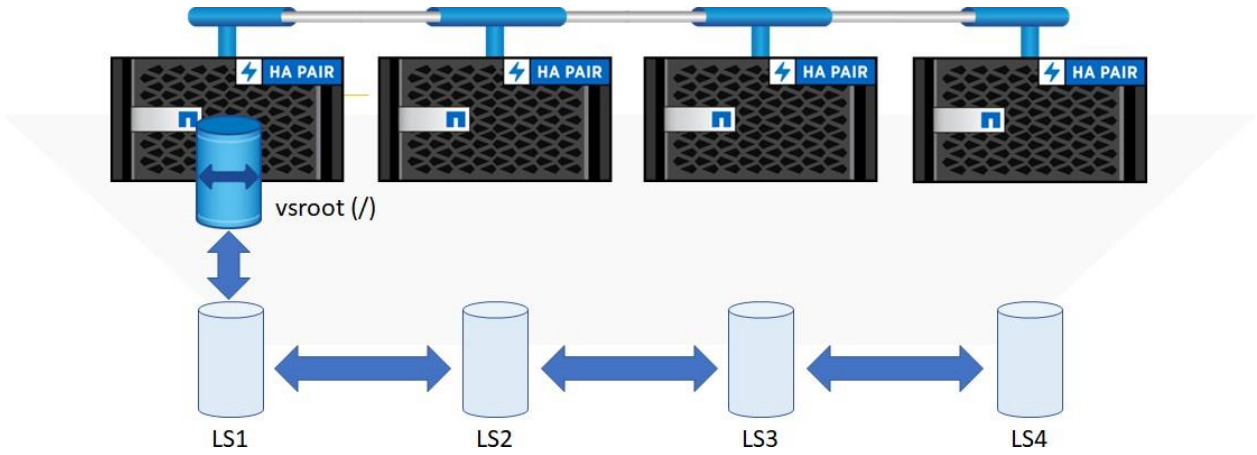
詳細については、負荷共有ミラー関係の作成と初期化を参照してください。

NetApp環境では、vsrootボリュームの負荷共有ミラー関係を作成することを強く推奨します。

注： NFSv4.xクライアントでは、NFSv4.xプロトコルの性質上、負荷共有ミラーボリュームを使用してファイルシステムをトラバースすることはできません。

図2 は、vsrootが使用できない場合に負荷共有ミラーがへのアクセスを提供する方法を示しています。

図2）vsrootボリュームの負荷共有ミラー保護



vsrootボリュームの負荷共有ミラーを作成するには、次の手順を実行します。

- 通常、vsrootボリュームのサイズは1GBです。新しいボリュームを作成する前にvsrootボリュームのサイズを確認し、新しいボリュームのサイズがすべて同じであることを確認してください。
- クラスタ内の各ノードでvsrootをミラーリングするデスティネーションボリュームを作成します。たとえば、4ノードクラスタでは、タイプがDPの新しいボリュームを4つ作成します。
- vsrootソースから、作成した新しいDPボリュームごとに新しいSnapMirror関係を作成します。ネームスペースルートの変更率に応じて、更新のスケジュールを指定します。たとえば、新しいボリュームを定期的に作成する場合は毎時、作成しない場合は毎日です。
- `initialize-ls-set` コマンドを使用してSnapMirrorを初期化します。

疑似ファイルシステム

ONTAPアーキテクチャにより、RFC 7530 NFSv4標準に準拠した真の疑似ファイルシステムを構築することが可能になります。

```
Servers that limit NFS access to "shares" or "exported" file systems should provide a pseudo-file
system into which the exported file systems can be integrated, so that clients can browse the
server's namespace. The clients' view of a pseudo-file system will be limited to paths that lead
to exported file systems.
```

セクション7.3：

```
NFSv4 servers avoid this namespace inconsistency by presenting all the exports within the
framework of a single-server namespace. An NFSv4 client uses LOOKUP and READDIR operations to
browse seamlessly from one export to another. Portions of the server namespace that are not
exported are bridged via a "pseudo-file system" that provides a view of exported directories
only. A pseudo-file system has a unique fsid and behaves like a normal, read-only file system.
```

ONTAPでは /vol、ONTAP 7-Modeでサポートされていたエクスポートボリュームに対する要件が廃止され、より標準化されたアプローチで疑似ファイルシステムが使用されるようになりました。これにより、///vol/vol07-Modeではのリダイレクタではなくが機能するため、既存のNFSインフラをNetAppストレージとシームレスに統合できるようになりました。

疑似ファイルシステムは、アクセス許可がより制限の厳しいものからより制限の低いものへと流れている場合にのみ、ONTAPで適用されます。たとえば、vsroot（にマウントされた /）のアクセス権がデータボリューム（など）のアクセス権よりも制限が厳しい場合は、/volname疑似ファイルシステムの概念が適用されます。

疑似ファイルシステムを使用すると、ストレージ管理者は、ジャンクションパスを使用して他のボリュームにボリュームをマウントすることで、必要に応じて独自のファイルシステムネームスペースを作成できます。この概念を 図1）クラスタネームスペースに示します。

疑似ファイルシステムと-actualのサポート

エクスポートオプションとしての-actualの使用はNetApp ONTAP 7-Modeでのみサポートされ、ONTAPではサポートされません。actualオプションは、ストレージ管理者がエクスポートパスをマスクして短縮名にリダイレクトする場合に使用します。

たとえば、ストレージ管理者がのパスを持っている場合、/dept1/folder1そのパスをとしてエクスポートできます /folder1。

actualがサポートされていない問題への対処

ほとんどの場合、-actual ONTAPではエクスポートオプションは必要ありません。オペレーティングシステムの設計は、エクスポートファイルで定義されたものではなく、自然な疑似ファイルシステムを可能にします。すべてが、にマウントされているSVMルートボリュームの下にマウントされ /ます。エクスポートはボリューム レベルまたはqtreeレベルで設定できるほか、複数のレベルまでジャンクションでき、実際のボリューム名とは関係のない名前を付けることができます。

表5）エクスポートの例

エクスポートパス	エクスポートされたオブジェクト
/vol1	vol1という名前のボリューム
/NFSvol	vol2という名前のボリューム
/vol1/NFSvol	vol1にジャンクションされたvol2という名前のボリューム
/vol1/qtree	vol1を親ボリュームとするqtreeという名前のqtree
/vol1/NFSvol/qtree1	vol1にジャンクションされたNFSvolを親ボリュームとするqtree1という名前のqtree

-actual ONTAP NFSアーキテクチャでは本来カバーされていないユースケースの1つに、-actual qtreeまたはフォルダがあります。たとえば、ストレージ管理者がqtreeやフォルダをなどのパスにエクスポートする場合、/folder1SVMのNFSエクスポートを使用してそのままエクスポートすることはできません。パスは代わりにです /volume/folder1。

7-Modeからのエクスポート例

```
/qtree -actual=/vol/vol1/qtree,rw,sec=sys
```

ONTAPでは、NFSクライアントがマウントするqtreeのパスは、ネームスペースでマウントされているパスと同じです。このパスが適切でない場合は、シンボリック リンクを利用してqtreeへのパスをマスクすることで対応します。

シンボリックリンクとは

symlinkはシンボリックリンクの省略形です。シンボリックリンクは、絶対パスまたは相対パスの形式で別のファイルまたはディレクトリへの参照を含む特別なタイプのファイルです。シンボリック リンクはクライアントには透過的で、データへの実際のパスとして機能します。

相対パスと絶対パス

シンボリック リンクには、相対パスまたは絶対パスを指定できます。絶対パスは、現在の作業ディレクトリや結合パスに関係なく、1つのファイルシステムの同じ場所を参照するパスです。相対パスは、作業ディレクトリを基準としたパスです。

たとえば、ユーザが/directoryのディレクトリ内にいて、/directory/userに移動する場合、そのユーザは相対パスを使用できます。

```
# cd user/
# pwd
/directory/user
```

または、絶対パスを使用することもできます。

```
# cd /directory/user
# pwd
/directory/user
```

NFSを使用してフォルダをマウントする場合は、シンボリックリンクに相対パスを使用することを推奨します。すべてのユーザがすべてのクライアントの同じマウントポイントにマウントする保証はないためです。相対パスを使用することで、絶対パスに関係なく機能するシンボリック リンクを作成できます。

シンボリックリンクを使用したシミュレーション-actual support

ONTAPでは、シンボリックリンクを使用して、-actual 7-Modeのエクスポートオプションと同じ動作をシミュレートできます。

たとえば、クラスタ内にqtreeが存在する場合、パスは次のようになります。

```
cluster::> qtree show -vserver flexvol -volume unix2 -qtree nfstree

                      Vserver Name: flexvol
                       Volume Name: unix2
                        Qtree Name: nfstree
                        Qtree Path: /vol/unix2/nfstree
                    Security Style: unix
                       Oplock Mode: enable
                  Unix Permissions: ---rwxr-xr-x
                          Qtree Id: 1
                      Qtree Status: normal
                     Export Policy: volume
        Is Export Policy Inherited: true
```

親ボリュームはで unix2 (/unix/unix2)、ボリュームにマウントされ unix (/unix)、vsroot（/）にマウントされます。

```
cluster::> vol show -vserver flexvol -volume unix2 -fields junction-path
  (volume show)
vserver volume junction-path
------- ------ ---------------
flexvol unix2 /unix/unix2
```

エクスポートされたパスは /parent_volume_path/qtree、 /vol/parent_volume_path/qtree 前述のではなくになります。showmount -e NFSクライアントからのコマンドの出力を次に示します。

```
/unix/unix2/nfstree                                                    (everyone)
```

/unix/unix2/nfstreeクライアントがパスの他の部分にアクセスできるようにするため、のパス全体を公開したくないストレージ管理者もいます。NFSクライアントへのこのパスをマスクするには、シンボリック リンク ボリュームまたはシンボリック リンク フォルダを作成し、ジャンクション パスにマウントします。例：

```
cluster::> vol create -vserver flexvol -volume symlinks -aggregate aggr1 -size 20m -state online
-security-style unix -junction-path /NFS_links
```

ボリューム サイズは小さく設定できますが（最小20MB）、ボリューム内のシンボリック リンクの数によります。各シンボリックリンクのサイズは4Kであるため、シンボリックリンクの数に対応するために、より大きなボリュームサイズを作成しなければならない場合があります。あるいは、vsrootの下にシンボリック リンク用のフォルダを作成します。

ボリュームまたはフォルダを作成したら、vsrootをNFSクライアントにマウントしてシンボリックリンクを作成します。

```
# mount -o nfsvers=3 x.x.x.e:/ /symlink
# mount | grep symlink
x.x.x.e:/ on /symlink type nfs (rw,nfsvers=3,addr=x.x.x.e)
```

注： vsrootの下のディレクトリを使用する場合は、vsrootをマウントしてディレクトリを作成します。

```
# mount -o nfsvers=3 x.x.x.e:/ /symlink
# mount | grep symlink
x.x.x.e:/ on /symlink type nfs (rw,nfsvers=3,addr=x.x.x.e)
# mkdir /symlink/symlinks
# ls -la /symlink | grep symlinks
drwxr-xr-x. 2 root root 4096 Apr 30 10:45 symlinks
```

qtreeへのシンボリックリンクを作成するには、-s オプション（s = symbolic）を使用します。リンク パスには、正確なパスを指定しなくてもシンボリック リンクを正しい場所に転送する相対パスを含める必要があります。リンクが目的のパスに移動できないフォルダ内にある場合は、パスに「../」を追加する必要があります。

たとえば、NFS_linksというフォルダがの下に作成され /、unixというボリュームもの下にマウントされている場合 /、/NFS_links シンボリックリンク原因に移動して作成します。親フォルダへのリダイレクトを必要とする相対パスです。

/NFS_linksにマウントされたシンボリックリンクボリューム内にシンボリックリンクを作成する例：

```
# mount -o nfsvers=3 x.x.x.e:/ /symlink/
# mount | grep symlink
x.x.x.e:/ on /symlink type nfs (rw,nfsvers=3,addr=x.x.x.e)
# cd /symlink/NFS_links
# pwd
/symlink/NFS_links
# ln -s ../unix/unix2/nfstree LINK
# ls -la /symlink/unix/unix2/nfstree/
total 8
drwxr-xr-x. 2 root root 4096 May 15 14:34 .
drwxr-xr-x. 3 root root 4096 Apr 29 16:47 ..
-rw-r--r--. 1 root root    0 May 15 14:34 you_are_here
# cd LINK
# ls -la
total 8
drwxr-xr-x. 2 root root 4096 May 15 14:34 .
drwxr-xr-x. 3 root root 4096 Apr 29 16:47 ..
-rw-r--r--. 1 root root    0 May 15 14:34 you_are_here
# pwd
```

```
/symlink/NFS_links/LINK
```

注： シンボリックリンクはの実際のパスを参照していますが、/unix/unix2/nfstreeは pwd シンボリックリンクのパス（）を返します /symlink/NFS_links/LINK。ファイルの you_are_here 日付とタイムスタンプは両方のパスで同じです。

パスには「../」が含まれているため、このシンボリック リンクは直接マウントできません。

vsrootにシンボリックリンクを作成する例：

```
# mount -o nfsvers=3 x.x.x.e:/ /symlink/
# mount | grep symlink
x.x.x.e:/ on /symlink type nfs (rw,nfsvers=3,addr=x.x.x.e)
# cd /symlink/
# pwd
/symlink
# ln -s unix/unix2/nfstree LINK1
# ls -la /symlink/unix/unix2/nfstree/
total 8
drwxr-xr-x. 2 root root 4096 May 15 14:34 .
drwxr-xr-x. 3 root root 4096 Apr 29 16:47 ..
-rw-r--r--. 1 root root    0 May 15 14:34 you_are_here
# cd LINK1
# ls -la
total 8
drwxr-xr-x. 2 root root 4096 May 15 14:34 .
drwxr-xr-x. 3 root root 4096 Apr 29 16:47 ..
-rw-r--r--. 1 root root    0 May 15 14:34 you_are_here
# pwd
/symlink/LINK1
```

繰り返しますが、実際のパスがであるという事実にもかかわらず、 /unix/unix2/nfstreeのあいまいなパスが表示されます/symlink/LINK1。ファイルの you_are_here 日付とタイムスタンプは両方のパスで同じです。さらに、vsrootパスの代わりに作成したシンボリック リンクをマウントすれば、エクスポート パスをさらに曖昧にすることができます。

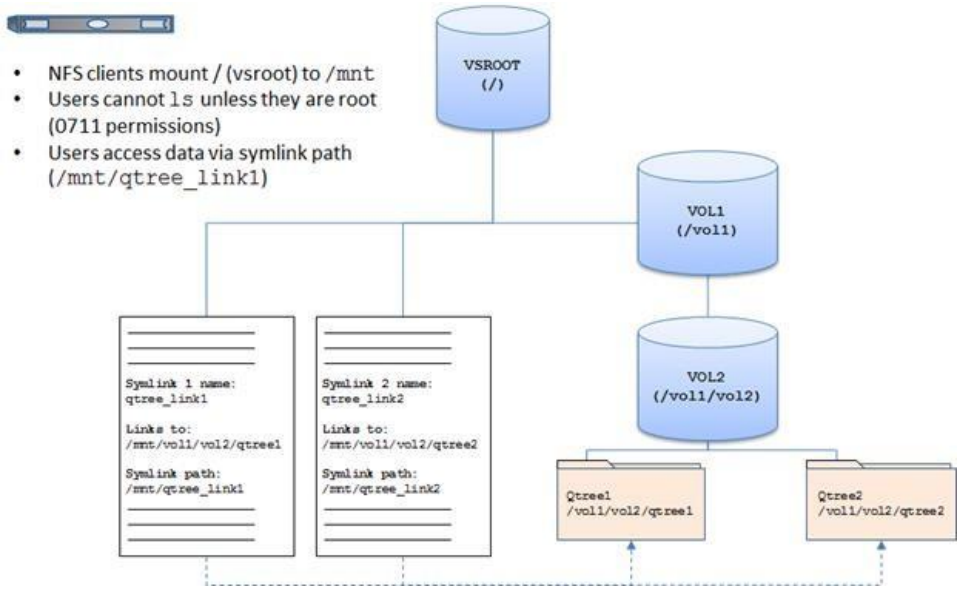
```
# mount -o nfsvers=3 x.x.x.e:/LINK1 /mnt
# mount | grep mnt
x.x.x.e:/LINK1 on /mnt type nfs (rw,nfsvers=3,addr=x.x.x.e
# cd /mnt
# pwd
/mnt
```

この設定を使用するユースケースの1つが自動マウントです。すべてのクライアントが同じパスをマウントできますが、そのパスがディレクトリ構造のどこに位置するかはクライアントにはわかりません。If clients mount the SVM root volume (/)を使用する場合は'必ず非管理クライアントに対してボリュームをロックダウンしてください.

ボリュームをロックダウンしてファイルやフォルダが表示されないようにする方法の詳細については、本ドキュメントの「SVMルートボリュームへのアクセスの制限」を参照してください。

図3 に、ネームスペースを作成してNAS処理用にパスをシンボリックリンクで曖昧にする方法の例を示します。

図3）vsrootを使用したシンボリックリンクの例



注：エクスポートポリシーとルールは、ボリュームとqtreeには適用できますが、フォルダまたはシンボリックリンクには適用できません。マウント ポイントとして使用するシンボリック リンクを作成する場合は、この点を考慮する必要があります。シンボリックリンクは、そのシンボリックリンクが配置されている親ボリュームのエクスポートポリシールールを継承します。

NetApp FlexGroupボリュウム

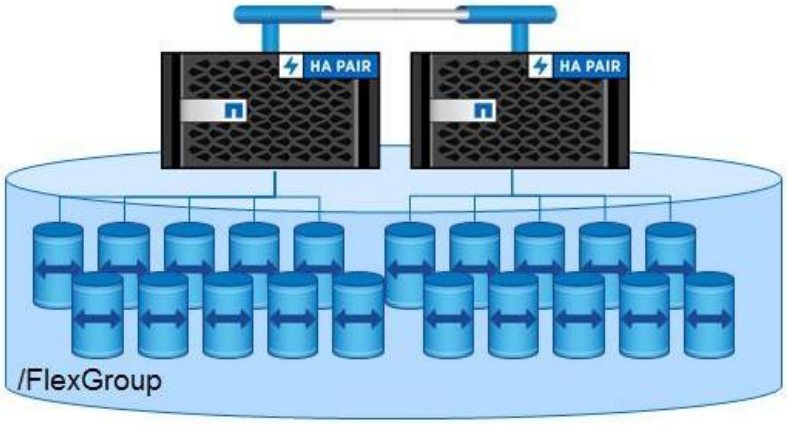
ONTAP 9.1以降では、NAS（CIFS / SMBおよびNFS）クライアントにストレージを提供する新しい方法として、NetApp FlexGroupボリュームが導入されました。

FlexGroupボリュームには次の利点があります。

- 単一のマウントポイントに対応する大容量（最大20PB、4、000億ファイルまで対応）
- 取り込み処理の同時実行により、FlexVolに比べてパフォーマンスが向上
- 導入と管理が容易

NetApp FlexGroupボリュームは、NFSクライアントに真のグローバルネームスペースを提供します。これは、クラスタ内の複数のノードにまたがる単一の大規模なストレージバケットであり、NASワークロードに対してメタデータ処理を並行して実行します。

図4）NetApp FlexGroupボリューム



理想的なユースケース

FlexGroupボリュームは、取り込み時の負荷が高く（新規データの作成）、同時処理が多く、サブディレクトリ間で均等に分散されている次のようなワークロードに最適です。

- EDA（電子設計自動化）
- ログ ファイルのリポジトリ
- ソフトウェアのビルド / テスト環境（GITなど）
- 地殻解析/石油/ガス
- メディア資産またはHIPAAアーカイブ
- ファイル ストリーミングのワークフロー
- 非構造化NASデータ（ホーム ディレクトリなど）
- ビッグデータ/データサイエンス
- 人工知能と機械学習
- ホーム ディレクトリ

FlexGroupボリュームの詳細については、次のリソースを参照してください。

- TR-4571：『NetApp FlexGroup Volume Best Practices and Implementation Guide』
- TR-4571-a：『Top Best Practices‐NetApp ONTAP FlexGroup Volumes』
- TR-4617：『Electronic Design Automation Best Practices』
- TR-4678：『Data Protection and Backup‐NetApp ONTAP FlexGroup Volumes』

NetApp FlexCacheホリユウム

ONTAP 9.5では、読み取り負荷の高いNFSワークロードをNetApp FlexCacheボリュームと呼ばれる複数のノードに分散する新機能が導入されています。

ONTAPのFlexCacheは、リモートの場所にあるボリュームの書き込み可能な永続的キャッシュを提供します。キャッシュは、ホストとデータソースの間にある一時的なストレージの場所です。キャッシュの目的は、ソースデータからデータをフェッチするよりも高速にデータを提供できるように、ソースデータの頻繁にアクセスされる部分を格納することです。キャッシュは、データが複数回アクセスされ、複数のホストで共有される読み取り処理の多い環境で役立ちます。キャッシュでは、次の2つの方法のいずれかを使用してデータを迅速に提供できます。

- キャッシュシステムの方が、データソースを使用するシステムよりも高速です。これは、キャッシュ内のストレージの高速化（SSDとHDDの高速化）、キャッシュ内の処理能力の向上、キャッシュ内のメモリの高速化（または高速化）によって実現できます。
- キャッシュ用のストレージスペースはホストに物理的に近いため、データに到達するまでにそれほど時間はかかりません。

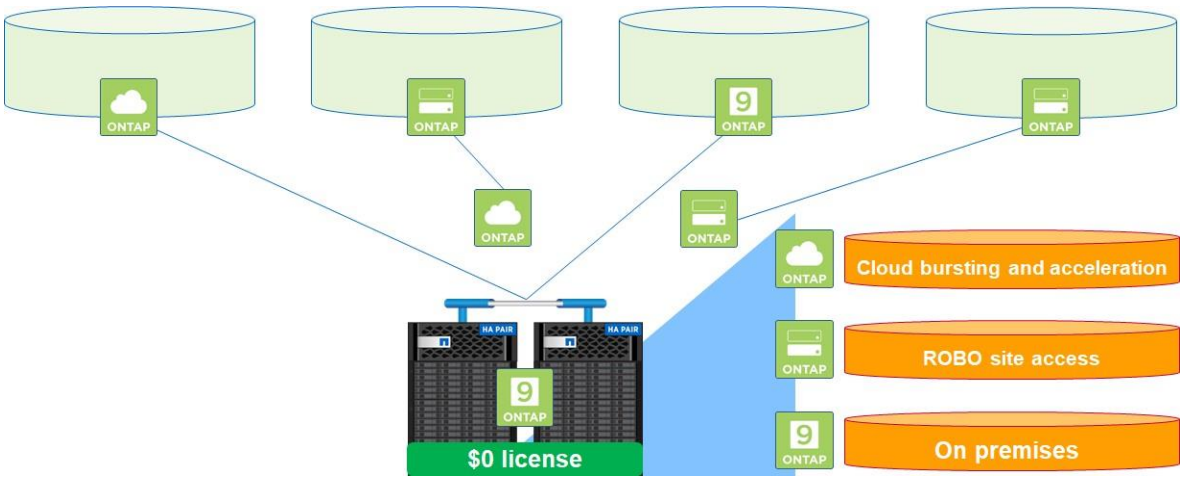
キャッシュは、さまざまなアーキテクチャ、ポリシー、セマンティクスで実装されるため、データがキャッシュに保存されてホストに提供されるときにデータの整合性が保護されます。

FlexCacheには次のようなメリットがあります。

- 負荷分散によるパフォーマンスの向上
- クライアントアクセスポイントの近くにデータを配置することでレイテンシを低減
- ネットワーク切断時にキャッシュされたデータを提供することで可用性を向上

FlexCacheは、キャッシュの一貫性、データの整合性、ストレージの効率的な使用をスケーラブルで高性能な方法で維持しながら、上記のすべての利点を提供します。

図5）NetApp FlexCacheボリューム



FlexCacheはスパースコピーです。元のデータセットのすべてのファイルをキャッシュできるわけではありません。キャッシュされたinodeのすべてのデータブロックがキャッシュに存在できるわけではありません。作業データセット（最近使用したデータ）の保持に優先順位を付け、ストレージを効率的に使用します。

FlexCacheを使用すると、ディザスタリカバリやその他の企業データ戦略の管理をオリジンに実装するだけで済みます。データ管理はソース上でのみ行われるため、FlexCacheを使用すると、リソースの使用効率が向上し、データ管理とディザスタリカバリ戦略が簡易化されます。

FlexGroupボリュームの詳細については、次のリソースを参照してください。

- TR-4743：『FlexCache in ONTAP』

ファイルロックの概念

ファイルロックは、ファイルを開いて使用しているときに、そのファイルが現在ロックされていることを他のクライアントに通知することで、ファイルの整合性を維持する方法です。NFSでは、ファイルロックメカニズムは使用するNFSのバージョンによって異なります。

NFSv3ロック

NFSv3は、Network Lock Manager（NLM；ネットワークロックマネージャ）やNetwork Status Monitor（NSM；ネットワークステータスモニタ）などの補助プロトコルを使用して、NFSクライアントとサーバ間でファイルロックを調整します。NLMはロックの確立と解放に役立ち、NSMはサーバのリブートをピアに通知します。NFSv3ロックでは、クライアントのリブート時にサーバでロックを解除する必要があります。サーバがリブートすると、クライアントはサーバに保持されているロックを通知します。場合によっては、ロックメカニズムが適切に通信できず、古いロックがサーバに残っているため、手動でクリアする必要があります。

NFSv4.xロック

NFSv4.xでは、NFSプロトコルに統合されたリースベースのロックモデルが使用されます。つまり、維持したり心配したりする補助サービスはなく、すべてのロックはNFSv4.x通信でカプセル化されます。

サーバまたはクライアントをリブートしたときに、指定した猶予期間中にロックを再確立できないと、ロックは期限切れになります。ONTAP NFSサーバは `-v4-grace- seconds`、オプションとを使用してこのロックタイムアウト時間を制御します `-v4-lease-seconds`。

- `-v4-lease-seconds` クライアントがリースを更新するまでにリースが許可される期間を示します。デフォルトは30秒です。最小値は10秒で、最大値はの値の-1秒 `-v4-grace-seconds`です。
- `-v4-grace-seconds` ノードのリブート時（フェイルオーバーやギブバック時など）にクライアントがONTAPからロックの再要求を試みる時間。デフォルトは45秒で、 `-v4-lease-seconds` 値の+1秒と最大90秒の範囲で変更できます。

まれに、lease secondsの値に指定された速度でロックが解放されず、2つのリース期間にわたってロックが解放されることがあります。たとえば、grace secondsが45秒に設定されている場合、ロックを解除するには90秒かかることがあります。詳細については、バグ957529を参照してください。これらの値がストレージフェイルオーバーに与える影響については、「NFSv4.xのロックがフェイルオーバーに及ぼす影響」を参照してください。

マルチプロトコルNASのロックの動作

マルチプロトコルNAS環境でファイルロックを使用する場合は、使用しているNASプロトコルによって動作が異なります。

- NASクライアントがSMBの場合、ファイルロックは必須ロックです。
- NASクライアントがNFSの場合、ファイルロックはアドバイザリロックです。

これが意味すること

NFSファイルとSMBファイルのロックの違いのため、SMBアプリケーションですでに開いているファイルにNFSクライアントからアクセスすると、エラーになる場合があります。

NFSクライアントがSMBアプリケーションによってロックされたファイルにアクセスすると、次のいずれかの状態になります。

- mixed形式またはNTFS形式のボリュームでは `rm`、、などのファイル操作で `rmdirmv` NFSアプリケーションを原因できない場合があります。
- NFSの読み取りと書き込みの処理は、SMBの読み取り拒否および書き込み拒否のオープン モードによってそれぞれ拒否されます。
- また、ファイルの書き込み対象となる範囲が、排他的なSMBバイトロックでロックされている場合も、NFSの書き込みの処理はエラーになります。UNIXセキュリティ形式のボリュームでは、NFSのリンク解除および名前変更処理でSMBのロック状態が無視され、ファイルへのアクセスが許可されます。UNIXセキュリティ形式のボリューム上の他のすべてのNFS処理では、SMBロック状態が維持されます。

マルチプロトコルNASの詳細については、「マルチプロトコルNAS」を参照してください。

ロックの種類

NFSロックには、次のようないくつかの種類があります。

- **共有ロック。**共有ロックは複数のプロセスで同時に使用でき、ファイルに排他ロックがない場合にのみ発行できます。これらは読み取り専用の処理を目的としていますが、書き込み（データベースなど）に使用できます。
- **排他ロック。**これらは、CIFS / SMBの排他ロックと同じように動作します。排他ロックが設定されている場合は、ファイルを使用できるプロセスは1つだけです。他のプロセスがファイルをロックした場合、そのプロセスがフォークされていない限り、排他ロックは発行できません。
- **委譲。**委譲はNFSv4.xでのみ使用され、NFSサーバオプションが有効になっていて、クライアントがNFSv4.xの委譲をサポートしている場合に割り当てられます。委譲は、クライアントが使用しているファイルに対する「ソフト」ロックを作成することで、クライアント側で処理をキャッシュする手段を提供します。これは、SMBの便宜的ロックと同様に、クライアントとサーバの間で実行される呼び出しの数を減らすことで、処理のパフォーマンスの一部の側面を向上させるのに役立ちます。委譲の詳細については、「NFSv4.1の委譲」を参照してください。
- **バイト範囲ロック。**バイト範囲ロックは、ファイル全体をロックするのではなく、ファイルの一部だけをロックします。

  メモ： ロックの動作は、ロックの種類、クライアントのオペレーティングシステムのバージョン、および使用されているNFSのバージョンによって異なります。想定される動作を測定するために、環境内のロックをテストしてください。

ONTAPでのファイルロックの詳細については、製品ドキュメントの「ファイルロックの管理」を参照してください。

クライアントでの手動ロックの確立

NFSロックをテストするには、クライアントがNFSサーバにロックを確立するように指示する必要があります。ただし、すべてのアプリケーションがロックを使用するわけではありません。たとえば、「vi」のようなアプリケーションはファイルをロックしません。代わりに、非表示のスワップファイルを同じフォルダに作成し、アプリケーションを閉じたときにそのファイルへの書き込みをコミットします。その後、古いファイルが削除され、スワップファイルの名前がファイル名に変更されます。

ただし、ロックを手動で確立するためのユーティリティがあります。たとえば、flockはファイルをロックできます。

1. ファイルのロックを確立するには、まずを実行し `exec` て数値IDを割り当てます。

```
# exec 4<>v4user_file
```

2. flockを使用して、ファイルに共有ロックまたは排他ロックを作成します。

```
# flock

Usage:
 flock [options] <file|directory> <command> [command args]
 flock [options] <file|directory> -c <command>
 flock [options] <file descriptor number>

Options:
 -s --shared              get a shared lock
 -x --exclusive           get an exclusive lock (default)
 -u --unlock              remove a lock
 -n --nonblock            fail rather than wait
 -w --timeout <secs>      wait for a limited amount of time
 -E --conflict-exit-code <number> exit code after conflict or timeout
 -o --close               close file descriptor before running command
 -c --command <command> run a single command string through the shell

 -h, --help     display this help and exit
 -V, --version output version information and exit

# flock -n 4
```

3. ONTAP SVMでロックされていることを確認します。

```
cluster::*> vserver locks show -vserver DEMO

Notice: Using this command can impact system performance. It is recommended
that you specify both the vserver and the volume when issuing this command to
minimize the scope of the command's operation. To abort the command, press Ctrl-C.

Vserver:  DEMO
Volume   Object Path                LIF          Protocol Lock Type    Client
-------- --------------------------- ------------ -------- ------------ ---------
home     /home/v4user_file           data2        nlm      byte-range 10.x.x.x
               Bytelock Offset(Length): 0 (18446744073709551615)
```

4. ファイルのロックを解除します。

```
# flock -u -n 4
```

注： ファイルを手動でロックすると、ファイルのオープンと編集の操作をテストしたり、ファイルロックでストレージフェイルオーバーイベントがどのように処理されるかを確認したりできます。

エクスポートの概念

ONTAP内のボリュームは、あるクライアントまたは一連のクライアントからアクセス可能なパスをエクスポートすることで、NFSクライアントと共有されます。ボリュームがSVMのネームスペースにマウントされると、ファイルハンドルが作成され、mountコマンドで要求されたときにNFSクライアントに提供されます。エクスポートに対する権限は、ストレージ管理者が設定できるエクスポートポリシーとルールによって定義されます。

エクスポート ポリシーとルールの概念

ONTAPは、セキュリティを制御するエクスポートポリシールールのコンテナとしてエクスポートポリシーを提供します。これらのポリシーはレプリケートされたデータベースに格納されるため、単一のノードに分離されるのではなく、クラスタ内のすべてのノードでエクスポートを使用できます。

これらのボリュームへのNFSアクセスを提供または制限するために、エクスポートポリシールールが作成されます。これらのルールでは、読み取り、書き込み、ルートアクセスを定義したり、クライアントリストを指定したりできます。1つのポリシーに複数のルールを含めることができ、1つのルールに複数のクライアントを含めることができます。

デフォルトのエクスポートポリシー

新しく作成したSVMには、defaultという名前のエクスポートポリシーが含まれています。このエクスポートポリシーは、名前変更や修正はできますが、削除はできません。NFSサーバを作成すると、デフォルトのポリシーが自動的に作成されてvsrootボリュームに適用されます。ただし、デフォルトポリシーにはエクスポートルールは含まれないため、デフォルトのエクスポートポリシーを使用するボリュームにはルールを追加するまでアクセスできません。エクスポートポリシーが定義されていない場合は、新しいボリュームの作成時にvsrootボリュームのエクスポートポリシーが継承されます。

VSrootとボリュームのトラバーサル

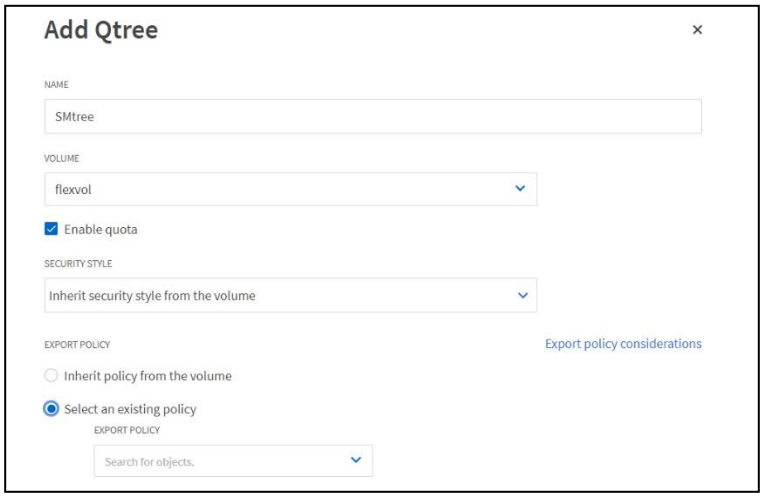
エクスポートポリシーはデフォルトで継承されるため、NetAppでは、ルールの割り当て時にSVMのルートボリューム（vsroot）への読み取りアクセスをNFSクライアントに許可することを推奨しています。「default」エクスポートポリシーにvsrootへの読み取りアクセスを制限するルールを設定すると、そのSVMの下に作成されたボリュームへのトラバーサルが拒否され、原因のマウントが失敗します。これは、vsrootが「/junction」へのパスの「/」にあたり、マウントおよびトラバースの可否を左右するためです。

qtreeエクスポート

ONTAPでは、ボリュームおよび基盤となるqtreeに対してエクスポートポリシーとルールを設定できます。これにより、ONTAP内のストレージ管理ディレクトリへのクライアントアクセスを制限または許可し、ストレージ管理者がホームディレクトリなどのワークロードをより簡単に管理できるようになります。

デフォルトでは、qtreeは親ボリュームのエクスポートポリシーを継承します。ONTAP System Managerでqtreeを作成するとき、または `-export- policy` CLIオプションを使用すると、エクスポートポリシーとルールを明示的に選択または作成できます。

図6）qtreeエクスポート仕様–ONTAP System Manager



qtree IDに関する考慮事項

継承されていないエクスポートポリシーをqtreeに適用すると、qtree間の操作を処理する際にNFSファイルハンドルがわずかに変更されます。詳細については、qtree IDおよび名前変更の動作を参照してください。

vsrootへノアクセスセイキヨ

vsrootへの読み取り / 書き込みアクセスを制御するには、ボリュームのUNIX権限、ACL、またはその両方を使用します。ボリュームの所有者でないユーザには、vsrootへの書き込み権限を制限することを推奨します（最高でも0755）。

特に指定がないかぎり、ボリュームの作成時のデフォルト値は次のとおりです。

- 0755は、ボリュームに設定されるデフォルトのUNIXセキュリティです。
- デフォルトの所有者はUID 0で、デフォルトのグループはGID 1です。

vsrootをトラバースして、をマウントできるNFSクライアントへの読み取り/リストアクセスも禁止するには、次の2つの方法があります。

オプション1：vsrootでUNIXモードビットをロックダウンする

vsrootをユーザにロックダウンする最も簡単な方法は、クラスタから所有権と権限を管理することです。

1. SVMに固有のローカルUNIXユーザを作成します。たとえば、SVM自体と同じ名前のUNIXユーザを指定できます。
2. vsrootボリュームを新しいUNIXユーザに設定します。ほとんどのNFSクライアントには「root」ユーザが設定されています。つまり、デフォルトでは、vsrootボリュームにrootユーザによるアクセスが多すぎる可能性があります。
3. グループやその他のユーザをトラバース権限のみに制限し、ボリュームの所有者に必要な権限を残しておくUNIX権限を使用します。（例：0611）

オプション2：NFSv4.xまたはNTFS ACLを使用してvsrootをロックダウンする

vsrootをロックダウンするもう1つの方法は、ACLを活用して、一部のユーザまたはグループを除くすべてのユーザに対して権限のトラバースを制限することです。これは、NFSv4.x ACLを使用して（NFSv3を使用してマウントする場合も含む）、またはCIFS / SMBプロトコルを提供する環境ではNTFS権限を使用して実行で

きます。NFSv3マウントでNFSv4.x ACLを使用する方法については、「NFSv3マウントでNFSv4.x ACLを使用する」を参照してください。

エクスポートポリシールール：オプション

このドキュメントの付録では、エクスポートポリシールールで使用されるさまざまなオプションとその用途、および例を示します。エクスポートポリシールールのほとんどのオプションは `export-policy rule show` 、コマンドまたはONTAP System Managerを使用して表示できます。

エクスポートポリシールール：継承

ONTAPでは、エクスポートポリシールールは適用先のボリュームとqtreeにのみ影響します。たとえば、SVMルートボリュームに、ルートアクセスを特定のクライアントまたは一部のクライアントに制限する制限的なエクスポートポリシールールがある場合、SVMルートボリューム（にマウント/）の下に存在するデータボリューム 適用されているエクスポートポリシーのみが適用されます。唯一の例外は、クライアントへの読み取りアクセスを拒否するエクスポートポリシールールがボリュームに設定されており、クライアントがそのパス内のボリュームをトラバースする必要がある場合です。現在、ONTAPには、NFSの「トラバースチェックのバイパス」という概念はありません。エクスポートポリシールールの継承の例については、「エクスポートポリシールールの継承の例」を参照してください。

エクスポートポリシールール：インデックス

ONTAPを使用すると、ストレージ管理者はエクスポートポリシールールの優先順位を設定して、特定の順序でルールが適用されるようにすることができます。ポリシーはアクセスが試みられたときに評価され、ルールは0～999999999の順に読み取られます。

注： ルールインデックス999999999は絶対最大値ですが、NetAppでは推奨していません。インデックスにはもっと実際的な数値を使用してください。

番号の小さいルール インデックス（1など）が読み取られてあるサブネットにアクセスが許可されたあとに、そのサブネット内のホストが番号の大きなインデックス（99など）のルールによってアクセスを拒否された場合、そのホストはポリシー内で先に読み取られたアクセスを許可するルールに基づいてアクセスを許可されます。

これとは逆に、クライアントがインデックスの番号の小さいエクスポート ポリシー ルールでアクセスを拒否されたあとに、ポリシー内の後続のグローバル エクスポート ポリシー ルール（clientmatch 0.0.0.0/0など）でアクセスを許可された場合、そのクライアントはアクセスを拒否されます。

ポリシールールのルールインデックスは `export-policy rule setindex` 、コマンドを使用するか、ONTAP System Managerで[Move Up/Move Down]を使用して並べ替えることができます。

図7 は、ONTAPシステムマネージャでのルールインデックスの並べ替えを示しています。

図7）ONTAPシステムマネージャでのルールインデックスの並べ替え

ONTAPでクライアントに許可されるアクセスと許可されないアクセスを決定する際には、エクスポートポリシールールの順序を考慮することが重要です。複数のエクスポート ポリシー ルールを使用する場合は、幅広いクライアントにアクセスを禁止 / 許可するルールによって、同じクライアントにアクセスを禁止 / 許可するルールが上書きされないようにしてください。ルールインデックスは、ルールが読み取られるときの順序を決定します。番号が大きいルールは、インデックス内の番号が小さいルールよりも優先されます。

注：より詳細なルール（管理ホストなどの特定のクライアント用など）を使用する場合は、ルールインデックスの上位に配置する必要があります。より広範なアクセスルールを低く設定する必要があります。たとえば、管理ホストのルールはルールインデックス1にあり、0.0.0.0/0のポリシーはインデックス99にあります。

エクスポートポリシールール：clientmatch

ストレージ管理者は、エクスポートポリシールールのclientmatchオプションを使用して、NFSエクスポートをマウントするためのアクセスリストを定義できます。また、クライアントがエクスポートをマウントできたあとにアクセス権限を大まかに制御する方法も定義できます。

NFSエクスポートポリシールールclientmatchの有効なエントリは次のとおりです。

- IP アドレス
- ホスト名
- ドメイン
- サブネット
- ネットグループ

  注：ONTAP 9.1以降では、複数のIPアドレスまたはホスト名をカンマで区切って1つのルールに定義できます。それぞれに固有のポリシールールを作成する必要はありません。

次の点を考慮する必要があります。

- clientmatchフィールドまたはネットグループにホスト名が使用されている場合は、ホスト名をIPアドレスに解決するために、稼働中のDNSサーバまたは手動ホストエントリが使用可能である必要があります。
- ネットグループを使用する場合は、ネットグループの先頭に@記号を付加して、ホスト名ではなくネットグループを指定していることをONTAPに通知する必要があります。
- 名前解決またはネットグループ検索にネームサービスを依存する場合は、必要なネームサービスにアクセスできるデータLIFがSVM内にあることを確認してください。
- ネームサービスの詳細については、TR-4668：『Name Services Best Practice Guide』を参照してください。

エクスポートポリシールール：キャッシュ

エクスポートポリシールール、クライアントホスト名、およびネットグループ情報はすべてONTAPにキャッシュされ、クラスタへの要求数が削減されます。これにより、要求のパフォーマンスが向上し、ネットワークやネームサービスサーバの負荷が軽減されます。

clientmatchキャッシュ

clientmatchエントリがキャッシュされると、そのエントリはSVMに対してローカルなままになり、キャッシュタイムアウト時間に達した場合やエクスポートポリシールールテーブルが変更された場合にフラッシュされます。デフォルトのキャッシュタイムアウト時間はONTAPのバージョンによって異なり、`export-policy access- cache config show` admin権限でコマンドを使用して確認できます。

デフォルト値は次のとおりです。

```
TTL For Positive Entries (Secs): 3600
TTL For Negative Entries (Secs): 3600
Harvest Timeout (Secs): 86400
```

エクスポートポリシーaccess-cache内の特定のクライアントを表示するには、次のadvanced権限のコマンドを使用します。

```
cluster::*> export-policy access-cache show -node node-02 -vserver NFS -policy default -address
x.x.x.x

                                        Node: node-02
                                     Vserver: NFS
                                  Policy Name: default
                                   IP Address: x.x.x.x
                       Access Cache Entry Flags: has-usable-data
                                  Result Code: 0
                  First Unresolved Rule Index: -
                       Unresolved Clientmatch: -
                Number of Matched Policy Rules: 1
           List of Matched Policy Rule Indexes: 2
                                Age of Entry: 11589s
                  Access Cache Entry Polarity: positive
Time Elapsed since Last Use for Access Check: 11298s
      Time Elapsed since Last Update Attempt: 11589s
                 Result of Last Update Attempt: 0
                  List of Client Match Strings: 0.0.0.0/0
```

ホスト名/ DNSキャッシュ

clientmatchにホスト名を設定すると、その名前がIPアドレスに解決されます。これは、SVMのネームサービススイッチ（ns-switch）で使用される順序に基づいて行われます。たとえば、ns-switchホストデータベースがfiles、dnsに設定されている場合、ONTAPはローカルホストファイルで一致するクライアントを検索してから、dnsを検索します。

名前検索後、ONTAPは結果をホストキャッシュにキャッシュします。このキャッシュの設定は変更可能で、advanced権限でONTAP CLIから照会およびフラッシュできます。

キャッシュを照会するには、次のコマンドを実行します。

```
cluster::*> name-service cache hosts forward-lookup show -vserver NFS
(vserver services name-service cache hosts forward-lookup show)
                    IP       Address IP                      Create
Vserver   Host      Protocol Family Address          Source Time         TTL(sec)
--------- ----------- ---------- ---------- ---------------- ---------- ------------ ------------

NFS       centos7.ntap.local
                    Any      Ipv4    x.x.x.x dns       3/26/2020 3600
                                                            16:31:11
```

ホストのキャッシュ設定を表示するには、次のコマンドを実行します。

```
cluster::*> name-service cache hosts settings show -vserver NFS -instance
  (vserver services name-service cache hosts settings show)

                    Vserver: NFS
         Is Cache Enabled?: true
Is Negative Cache Enabled?: true
              Time to Live: 24h
     Negative Time to Live: 1m
     Is TTL Taken from DNS: true
```

場合によっては、NFSクライアントのIPアドレスが変更されたときに、アクセスの問題を修正するためにhostsエントリのフラッシュが必要になることがあります。

ホストのキャッシュエントリをフラッシュするには、次のコマンドを実行します。

```
cluster::*> name-service cache hosts forward-lookup delete -vserver NFS ?
     -host      -protocol -sock-type -flags      -family
```

ネットグループキャッシング

clientmatchフィールドのネットグループをエクスポートルールに使用している場合、ONTAPはネットグループネームサービスサーバと通信してネットグループ情報を展開する追加の作業を行います。ns-switch内のネットグループデータベースは、ONTAPがネットグループを照会する順序を決定します。また、ネットグループのサポートにONTAPが使用する方法は、netgroup.byhostのサポートが有効か無効かによって異なります。netgroup.byhostの詳細については、TR-4835：『How to Configure LDAP in ONTAP』を参照してください。

- netgroup.byhostが無効になっている場合、ONTAPはネットグループ全体を照会し、すべてのネットグループエントリをキャッシュに取り込みます。ネットグループに数千のクライアントがある場合、そのプロセスが完了するまでに時間がかかることがあります。netgroup.byhostはデフォルトで無効になっています。
- netgroup.byhostが有効になっている場合、ONTAPはネームサービスに対してホストエントリと関連するネットグループマッピングのみを照会します。これにより、潜在的に数千のクライアントを検索する必要がないため、ネットグループのクエリに必要な時間が大幅に短縮されます。

これらのエントリは `vserver services name-service cache` 、コマンド内のネットグループキャッシュに追加されます。これらのキャッシュエントリは表示またはフラッシュでき、タイムアウト値を設定できます。

ネットグループキャッシュ設定を表示するには、次のコマンドを実行します。

```
cluster::*> name-service cache netgroups settings show -vserver NFS -instance
  (vserver services name-service cache netgroups settings show)

                    Vserver: NFS
         Is Cache Enabled?: true
Is Negative Cache Enabled?: true
              Time to Live: 24h
     Negative Time to Live: 1m
  TTL for netgroup members: 30m
```

ネットグループ全体がキャッシュされると、そのネットグループはメンバーキャッシュに配置されます。

```
cluster::*> name-service cache netgroups members show -vserver DEMO -netgroup netgroup1
  (vserver services name-service cache netgroups members show)

            Vserver: DEMO
           Netgroup: netgroup1
              Hosts: sles15-1,x.x.x.x
        Create Time: 3/26/2020 12:40:56
Source of the Entry: ldap
```

キャッシュされているネットグループエントリが1つだけの場合、IP-to-netgroupキャッシュおよびホストのリバースルックアップキャッシュにエントリが入力されます。

```
cluster::*> name-service cache netgroups ip-to-netgroup show -vserver DEMO -host x.x.x.y
  (vserver services name-service cache netgroups ip-to-netgroup show)
Vserver    IP Address Netgroup      Source Create Time
---------- ---------- ------------- ------ -----------
DEMO       x.x.x.y
                      netgroup1     ldap   3/26/2020 17:13:09

cluster::*> name-service cache hosts reverse-lookup show -vserver DEMO -ip x.x.x.y
  (vserver services name-service cache hosts reverse-lookup show)
Vserver    IP Address      Host                  Source Create Time     TTL(sec)
---------- --------------- --------------------- ------ -------------- ----------
DEMO       x.x.x.y   centos8-ipa.centos-ldap.local
                                                 dns    3/26/2020 17:13:09
                                                                       3600
```

キャッシュタイムアウトの変更に関する考慮事項

必要に応じて、キャッシュ設定を別の値に変更できます。

- タイムアウト値を大きくするとキャッシュエントリが長く保持されますが、クライアントがIPアドレスを変更した場合（クライアントIPアドレスにDHCPが使用されていてDNSが更新されない場合、エクスポートルールでIPアドレスが使用されている場合など）、クライアントアクセスの不整合が発生する可能性があります。
- タイムアウト値を小さくすると、キャッシュがフラッシュされる頻度が高くなり、より最新の情報が取得されますが、ネームサービスサーバへの負荷が増大し、クライアントからのマウント要求のレイテンシが増大する可能性があります。

ほとんどの場合、キャッシュタイムアウト値をそのままにしておくのが最善の方法です。詳細とガイダンスについては、TR-4668：『Name Services Best Practices』および TR-4835：『How to Configure LDAP in ONTAP』を参照してください。

exportfsのサポート

ONTAPでは、 `exportfs` は `export-policy` および `name-service` cacheコマンドに置き換えられています。を実行する `exportfs`と、次のように表示されます。

```
  "exportfs" is not supported: use the "vserver export-policy" command.
```

エクスポートポリシールール：アクセス検証

ONTAPにはコマンド（`export-policy check-access`）が用意されています。このコマンドを使用すると、エクスポートポリシーのアクセスルールセットをクライアントのアクセスと照合して、エクスポートポリシールールが導入前およびトラブルシューティングの際に適切に機能しているかどうかを判断できます。その機能は `exportfs -c` 機能に似ています。このコマンドは、NFSクライアントからの標準マウントで使用される、通常のネームサービス通信とキャッシュのやり取りをすべて利用します。

例 `export-policy check-access`：

```
cluster1::*> vserver export-policy check-access -vserver vs1 -client-ip 1.2.3.4 -volume flex_vol
-authentication-method sys -protocol nfs3 -access-type read

                                        Policy    Policy       Rule
Path                                    Policy    Owner     Owner Type  Index Access
--------------------------------------- --------- --------- ----------- ----------
/                                       default   vs1_root  volume          1 read
/dir1                                   default   vs1_root  volume          1 read
/dir1/dir2                              default   vs1_root  volume          1 read
/dir1/dir2/flex1                        data      flex_vol  volume         10 read
```

anonユーザ

匿名（anon）ユーザIDは、有効なNFSクレデンシャルのないクライアント要求にマッピングされるUNIXユーザIDまたはユーザ名を指定します。これにはrootユーザが含まれることがあります。ONTAPは、SVMで指定されたネームマッピング方式とネームスイッチ方式に照らしてユーザの有効なUIDを確認することで、ユーザのファイルアクセス権限を決定します。有効なUIDが特定されると、エクスポート ポリシー ルールに従ってそのUIDに許可するアクセスが決定されます。

`-anon` エクスポートポリシールールのオプションでは、有効なNFSクレデンシャルのないクライアント要求（rootユーザを含む）にマッピングされるUNIXユーザIDまたはユーザ名を指定できます。`-anon`エクスポートポリシールールの作成時に指定しなかった場合のデフォルト値は65534です。このUIDは、Linux環境では通常ユーザ名「nobody」または「nfsnobody」に関連付けられます。NetApp アプライアンスでは65534をユーザ「pcuser」（一般にマルチプロトコル処理に使用）として使用します。この違いのため、ローカル ファイルとNFSv4を使用する場合、65534にマップされたユーザの名前文字列が一致しなくなることがあります。この不一致は `/etc/idmapd.conf`、`/etc/default/nfs` 特に同じデータセットでマルチプロトコル（CIFSとNFS）を使用している場合に、クライアント（Linux）またはファイル（Solaris）のファイルに指定されたユーザとして原因ファイルが書き込まれる可能性があります。

rootユーザ

ONTAPでrootユーザを明示的に設定して共有へのルートアクセスを許可するマシンを指定するか、または

anon=0を指定する必要があります。-superuser ルートアクセスをさらに細かく制御する必要がある場合は、オプションを使用します。これらの設定が適切に構成されていないと、rootユーザ（0）としてNFS共有にアクセスしたときにPermission deniedが発生する可能性があります。-anon エクスポートポリシールールの作成時にこのオプションを指定しない場合、rootユーザIDはnobodyユーザにマッピングされます（65534）。

認証タイプ

AUTHタイプはNFSクライアントが認証する際に送信されます。AUTHタイプは、クライアントがサーバを認証する方法を指定するもので、クライアント側の設定に依存します。サポートされるAUTHタイプは以下のとおりです。

- **AUTH_NONE/AUTH_NULL**このAUTHタイプは、着信要求にIDがなく（NONEまたはNULL）、anonユーザにマッピングされることを示します。詳細については、『http://www.ietf.org/rfc/rfc1050.txt and http://www.ietf.org/rfc/rfc2623.txt』を参照してください。
- **AUTH_SYS / AUTH_UNIX。**このAUTHタイプは、ユーザがクライアント（またはシステム）で認証され、識別されたユーザとして受信されることを指定します。詳細については、『http://www.ietf.org/rfc/rfc1050.txt and http://www.ietf.org/rfc/rfc2623.txt』を参照してください。
- **AUTH_RPCGSS。**Kerberos対応のNFS認証です。

NFS共有へのルート アクセスを設定するには、いくつかの方法があります。例については、「rootユーザの制御例」を参照してください。

SVMルートボリュームへのアクセスの制限

デフォルトでは、SVMが作成されると、権限が755、owner:groupがroot（0）:root（0）のルート ボリュームが設定されます。これは次のことを意味します。

- ユーザroot（0）の有効な権限は7（フルコントロール）です。
- グループとその他の権限レベルは5（読み取りと実行）に設定されます。

この設定では、SVMルート ボリュームにアクセスするすべてのユーザが、SVMルート ボリューム（ジャンクション パスとして常に「/」にマウントされる）の下にマウントされるジャンクションを表示し、読み取ることができます。また、System Managerまたは vserver setup コマンドを使用してSVMを設定する際に作成されるデフォルトのエクスポートポリシールールは、SVMルートへのユーザアクセスを許可します。

SVMのセットアップで作成されるデフォルトのエクスポートポリシールールの例：

```
cluster::> export-policy rule show -vserver nfs_svm -policyname default -instance
  (vserver export-policy rule show)

                                Vserver: nfs_svm
                            Policy Name: default
                             Rule Index: 1
                        Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                          RO Access Rule: any
                          RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: none
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
```

上記のエクスポートポリシールールでは、すべてのクライアントにROとRWのいずれかのアクセスが許可されています。rootはanonに引き下げられ、anonは65534に設定されています。

たとえば、あるSVMに3つのデータボリュームがある場合、すべてがの下にマウントされ、 ls マウントにアクセスするすべてのユーザが基本的なコマンドを使用して表示できます。

```
# mount | grep /mnt
x.x.x.e:/ on /mnt type nfs (rw,nfsvers=3,addr=x.x.x.e)
# cd /mnt
# ls
nfs4 ntfs unix
```

特定のユーザ グループだけがデータ ボリュームを表示できるようにしたい場合、この動作は望ましくありません。ボリューム自体への読み取り / 書き込みアクセスは、権限とエクスポート ポリシー ルールを使用してデータ ボリューム単位で制限できますが、ユーザはデフォルトのポリシー ルールとボリューム権限を使用してその他のパスを表示できる状態です。

SVMルートボリュームの内容（および後続のデータボリュームパス）をユーザが表示できるように制限しながら、データアクセス用のジャンクションパスのトラバーサルを許可するには、次の手順を実行します。SVMルートボリュームを変更して、rootユーザにのみSVMルート内のフォルダのリストを許可することができます。そのためには、volume modifyコマンドを使用して、SVMルート ボリュームに対するUNIX権限を0711に変更します。

```
cluster::> volume modify -vserver nfs_svm -volume rootvol -unix-permissions 0711
```

削除後も、rootは所有者であるため、7つの権限を使用したフルコントロールがrootに付与されます。Groupやその他のユーザは、1モードビットごとにExecute権限を取得します。この権限では、cdを使用したパスのトラバースのみが許可されます。

rootユーザ以外のユーザが `ls`アクセスを試みた場合、そのユーザはアクセスを拒否されます。

```
sh-4.1$ ls
ls: cannot open directory .: Permission denied
```

多くの場合、NFSクライアントはrootユーザとしてワークステーションにログインします。System ManagerおよびSVMのセットアップで作成されるデフォルトのエクスポートポリシールールでは、ルートアクセスが制限されます。

```
# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
# ls -la
ls: cannot open directory .: Permission denied
```

これは、エクスポートポリシールールのsuperuser属性がNoneに設定されているためです。特定のクライアントがルートアクセスを必要とする場合は、ポリシーにエクスポートポリシールールを追加し、clientmatchフィールドにホストのIP、名前、ネットグループ、またはサブネットを指定することで制御できます。このルールを作成するときは、clientmatch 0.0.0.0/0や0/0（すべてのホスト）など、上書きされる可能性のあるルールの前にそのルールをリストします。

次に、管理ホストルールをポリシーに追加する例を示します。

```
cluster::> export-policy rule create -vserver nfs_svm -policyname default -clientmatch x.x.x.x -
rorule any -rwrule any -superuser any -ruleindex 1

cluster::> export-policy rule show -vserver nfs_svm -policyname default -ruleindex 1
  (vserver export-policy rule show)

                                Vserver: nfs_svm
                             Policy Name: default
                              Rule Index: 1
                          Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: x.x.x.x
                            RO Access Rule: any
                            RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: any
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true


cluster::> export-policy rule show -vserver nfs_svm -policyname default
  (vserver export-policy rule show)
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
---------------- ---------- ------ ---------- ------------------------- ------------
nfs_svm default          1       any      x.x.x.x        any
nfs_svm default          2       any      0.0.0.0/0              any
2 entries were displayed.
```

これで、クライアントはrootユーザとしてディレクトリを表示できるようになります。

```
# ifconfig | grep "inet addr"
          inet addr:x.x.x.x Bcast:x.x.225.255 Mask:255.255.255.0
          inet addr:127.0.0.1 Mask:255.0.0.0
# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
# ls
nfs4 ntfs unix
```

他のクライアントはrootとしてコンテンツを一覧表示できません。

```
# ifconfig | grep "inet addr"
          inet addr:x.x.x.y Bcast:x.x.225.255 Mask:255.255.255.0
# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
# mount | grep mnt
x.x.x.e:/ on /mnt type nfs (rw,nfsvers=3,addr=x.x.x.e)
# ls /mnt
ls: cannot open directory .: Permission denied
```

エクスポート ポリシー ルールの詳細およびそれらのルールがrootユーザに与える影響については、このドキュメントの「rootユーザ」セクションを参照してください。

モードビットの詳細については、次のリンクを参照してください。
http://www.zzee.com/solutions/unix-permissions.shtml

すべてのUIDを1つのUIDにマッピング（squash_all）

UNIXセキュリティ形式のボリュームにNFS経由でアクセスするユーザの一部またはすべてをどのUID（rootなど）にマッピングするかを、ストレージ管理者が制御したい場合があります。ボリュームのセキュリティ形式がNTFSである場合は、NFSサーバ オプションでデフォルトのWindowsユーザを設定するだけなので簡単です。一方、ボリュームのセキュリティ形式がUNIXである場合は、NFSクライアントからアクセスする際にネーム マッピングが実行されません。これを制御するには、エクスポート ポリシー ルールを作成します。

すべてのUIDを65534に引き下げる

次のエクスポートポリシールールの例は、特定のサブネットからシステムにアクセスするすべてのUID（rootを含む）に65534 UIDを使用するように設定する方法を示しています。このルールを使用して、ユーザがアクセスを制限するゲストアクセスポリシーを作成できます。そのためには、RO、RW、およびsuperuser認証タイプをNone、anon値を65534、clientmatchでサブネットを指定します。

```
                                    Vserver: nfs_svm
                                Policy Name: default
                                 Rule Index: 1
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: x.x.225.0/24
                             RO Access Rule: none
                             RW Access Rule: none
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: none
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
```

スヘテノUIDノrootニスル

次のエクスポート ポリシー ルールの例では、特定のサブネットからシステムにアクセスするすべてのUID（rootを含む）が、root（0）に関連付けられたUIDを使用するように設定しています。このルールを利用して特定のサブネットに属するユーザにフル アクセスを許可すると、権限管理の負荷を削減できます。このアクセスを有効にするには、ROとRWの認証タイプがNone、superuserの値がNone、anonの値が0、clientmatchの値がサブネットを指定します。

例：

```
                                    Vserver: nfs_svm
                                Policy Name: default
                                 Rule Index: 1
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: x.x.225.0/24
                             RO Access Rule: none
                             RW Access Rule: none
User ID To Which Anonymous Users Are Mapped: 0
                   Superuser Security Types: none
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true
```

特殊文字に関する考慮事項

Unicodeで最も一般的なテキスト文字（UTF-8形式でエンコードされている場合）は、3バイト以下のエンコードを使用します。この一般的なテキストには、中国語、日本語、ドイツ語など、現代のすべての文字言語が含まれています。しかし、 絵文字などの特殊文字の普及に伴い、UTF-8の一部の文字サイズが3バイトを超えています。たとえば、トロフィーシンボルはUTF-8エンコーディングで4バイトを必要とする文字です。

特殊文字には、次のものがあります。

- 絵文字
- 音楽記号
- 数学記号

FlexGroupボリュームに特殊文字が書き込まれると、次の動作が発生します。

```
# mkdir /flexgroup4TB/🏆
mkdir: cannot create directory ‘/flexgroup4TB/\360\237\217\206’: Permission denied
```

上記の例では、`\360\237\217\206` は `0xF0 0x9F 0x8F 0x86` トロフィーシンボルであるUTF-8の16進数です。

ONTAPソフトウェアでは、バグ229629に示すように、NFSで3バイトを超えるサイズのUTF-8がネイティブにサポートされていませんでした。3バイトを超える文字サイズを処理するために、ONTAPは余分なバイトをオペレーティングシステムのと呼ばれる領域に配置しまし `bagofbits`た。これらのビットは、クライアントが要求するまで保存されていました。次に、クライアントはrawビットの文字を解釈します。FlexVol `bagofbits`はすべてのONTAPリリースでサポートされ、FlexGroupボリュームは `bagofbits` ONTAP 9.2でのサポートを追加しました。

また、ONTAPには、`bagofbits` 処理の問題に関するイベント管理システムメッセージがあります。

```
Message Name: wafl.bagofbits.name
Severity: ERROR

Corrective Action: Use the "volume file show-inode" command with the file ID and volume name
information to find the file path. Access the parent directory from an NFSv3 client and rename
the entry using Unicode characters.

Description: This message occurs when a read directory request from an NFSv4 client is made to a
Unicode-based directory in which directory entries with no NFS alternate name contain non-Unicode
characters.
```

ボリューム言語utf8mb4のサポート

前述したように、特殊文字は、ネイティブでサポートされている3バイトのUTF-8エンコーディングを超える場合があります。その後、ONTAPはこの `bagofbits` 機能を使用して、これらの文字を使用できるようにします。

このinode情報の格納方法は理想的ではないため、ONTAP 9.5以降ではボリューム言語utf8mb4がサポートされるようになりました。ボリュームでこの言語を使用する場合、4バイトの特殊文字はにではなく適切に格納されます。`bagofbits`

ボリューム言語は、NFSv3クライアントから送信された名前をUnicodeに変換したり、ディスク上のUnicode名をNFSv3クライアントで想定されるエンコーディングに変換したりするために使用されます。UTF-8以外のエンコーディングを使用するようにNFSホストが設定されている従来の状況では、対応するボリューム言語を使用する必要があります。最近ではUTF-8の使用がほぼ一般的になっており、ボリューム言語はUTF-8である可能性があります。

NFSv4ではUTF-8を使用する必要があるため、NFSv4ホストでUTF-8以外のエンコードを使用する必要はありません。同様に、CIFSはUnicodeを標準で使用するため、任意のボリューム言語で動作します。ただし、Unicode名が基本平面より上にあるファイルはutf8mb4以外のボリュームでは正しく変換されないため、utf8mb4を使用することをお勧めします。

UTF-8フンシニカンスルモンタイ

文字表現に0x80（octal\0200）を含むUTF-8ファイル名をNFSマウントで管理できない場合があります。これらの文字の多くは、Unicode General句読点ブロックに含まれています。たとえば、名前`'test·file'`は`'test\xe2\x80\xa2file'`としてエンコードされており、UTF-8シーケンスに0x80が含まれているため、その名前が影響を受ける可能性があります。詳細については、バグ998468を参照してください。

注 ：この問題は、ONTAP 8.3.2より前に作成されたファイルにのみ影響します。

`-v3-search-unconverted-filename` この問題を回避するために、ONTAP 9でオプションが追加されました。

```
[-v3-search-unconverted-filename {enabled|disabled}] - Lookup for the filename in unconverted
language if converted language lookup fails (privilege: advanced)
This optional parameter specifies whether to continue the search with unconverted name while
doing lookup in a directory.
```

NFSのバージョンに関する考慮事項

NFSv2に関する考慮事項

NFSv2のサポートはONTAP 8.2で廃止され、使用できなくなりました。

NFSv3に関する考慮事項

次のセクションでは、ONTAPの機能、既知の問題、およびNFSv3に関する考慮事項について説明します。

NFSv3のマウント時の動作

NFSv3経由でファイルシステムをマウントする場合は、次の手順が実行されます。

1. NFSサーバのポート111（ポートマッパー）に対してリモート手順コール（RPC）が実行され、ポートマッパーを介したTCP接続が試行されます。
2. RPCの確認が完了すると、portmapperはNFSサーバデータLIFのポート111に対してGETPORT呼び出しを実行して、NFSが許可されているポートを取得します。
3. NFSサーバはポート2049（NFS）をクライアントに返します。
4. その後、クライアントはポート111への接続を閉じます。
5. NFSサーバデータLIFのポート2049への新しいRPC呼び出しが行われます。
6. NFSサーバから呼び出しが正常に返され、クライアントはNFSサーバのデータLIFのポート2049にNFS NULL呼び出しを送信します。親ボリュームのエクスポートポリシールールでマウントへのアクセスが許可されているかどうかがチェックされます。この場合、親ボリュームは、 /またはSVMルートにマウントされます。
7. NFS NULL呼び出しが正常に返され、クライアントはマウントの試行を続行します。

8. Portmapperは、NFSサーバのデータLIFに別のGETPORT呼び出しを送信して `mountd` ポートを要求し、クレデンシャル、NFSバージョン番号、およびマウントでTCPとUDPのどちらを使用するかを指定します。
9. クラスタはNFS設定をチェックし、指定したクレデンシャルでマウントが許可されているかどうかをエクスポート済みボリュームのエクスポートポリシールールに基づいて検証します。ネームサービスサーバ（DNS、NIS、LDAPなど）には必要に応じてアクセスします。キャッシュにデータが格納されている。NFSバージョンまたはTCP / UDPが許可されていない場合、クライアントはエラーを報告します。
10. 指定されたバージョンがサポートされていて、マウントで指定されたTCP接続またはUDP接続を使用できるかどうかは、NFSサーバが正常に応答します。また、AUTHセキュリティプロバイダーがサポートされている場合（AUTH_SYSやAUTH_GSSなど）も応答します。
11. GETPORTコールが通過すると、クライアントは `mount mountd`、コマンドで指定したジャンクションパスに対して、NFSサーバのデータLIFのポート635()を介してV3 mntコールを発行します。
12. ONTAPはエクスポートへの指定されたパスを検索します。エントリが存在する場合は、各ボリュームに固有のファイルハンドル情報がクラスタによって収集されます。
13. NFSサーバからクライアントにファイルハンドルが返され、エクスポートポリシールールでサポートされている認証タイプが返されます。サーバが提供する認証の種類がクライアントが送信したものと一致した場合、マウントは成功します。
14. クライアントからのポートマッパーは、次にNFSの別のGETPORT呼び出しを送信し、今度はクライアントのホスト名を提供します。
15. NFSサーバがポート2049で応答し、コールは成功します。
16. クライアントからNFSデータLIFのポート2049経由でもう1つのNFS NULL呼び出しが実行され、NFSサーバから確認応答が返されます。
17. fsinfoおよびpathconf情報を含む一連のNFSパケットが、クライアントとNFSサーバの間でやり取りされます。

ONTAPでのファイルシステムIDの変更による影響

NFSは、クライアントとサーバ間のやり取りにファイルシステムID（FSID）を使用します。NFSクライアントは、このFSIDによって、NFSサーバのファイルシステムのどこにデータがあるかを認識します。ONTAPでは、ジャンクション パスを使用することで複数のファイルシステムが複数のノードにまたがることが可能なため、データの場所に応じてFSIDが変わる可能性があります。一部の古いLinuxクライアントでは、FSIDの変更を区別できず、基本的な属性処理（ `chown` やなど）でエラーが発生することがあります `chmod`。

この問題の例は、バグ671319に記載されています。NFSv3でFSIDの変更を無効にする場合は `-v3-64bit-identifiers`、必ずONTAP 9でオプションを有効にしてください。ただし、このオプションは、32ビットのファイルIDを必要とする古いレガシーアプリケーションに影響する可能性があることに注意してください。

これがファイル数の多い環境に与える影響については、TR-4571：『NetApp FlexGroup Volume Best Practices and Implementation』を参照してください。

NetApp SnapshotコピーでのFSIDの動作

ボリュームのNetApp Snapshot™コピーが作成されると、遅延アクセスのためにファイルのinodeのコピーがファイルシステムに保持されます。理論上は、ファイルが2箇所に存在することになります。

NFSv3では、実質的に同じファイルに2つのコピーが存在しても、それらのファイルのFSIDは同一となりません。ファイルのFSIDは、NetApp WAFL inode番号、ボリュームID、およびSnapshotコピーIDを組み合わせて作成されます。SnapshotコピーのIDはそれぞれ異なるため、NFSv3では、`-v3-fsid- change` オプションの設定に関係なく、ファイルのすべてのSnapshotコピーのFSIDが異なります。NFS RFC仕様では、ファイルのバージョン間でFSIDが同一であることは要求されていません。

Storage Virtual Machineのディザスタリカバリでの FSIDの変更

ONTAP 8.3.1では、Storage Virtual Machineディザスタリカバリ（SVM DR）と呼ばれる、SVM全体のディザスタリカバリを可能にする新機能が導入されています。この機能については、TR-4015：『SnapMirrorの設定およびベストプラクティスガイド』で説明しています。

ONTAP 9.0より前のバージョンでNFSエクスポートでSVM DRを使用すると、エクスポートのFSIDが変更されるため、クライアントはエクスポートをデスティネーションシステムに再マウントする必要があります。それ以外の場合は `stale` 、それらのマウントでのNFS処理についてと表示されます。マウントのFSIDをSVM DR関係で保持する必要がある場合は `-is-msid-preserve` 、診断権限モードでオプションをTrueに設定してデスティネーションSVMを作成する必要があります。このオプションを設定すると、SVM DRで使用されているSnapMirror関係の `-msid- preserve snapmirror show` 出力にTrueと表示されます。SVM DRの更新は非同期であるため、この方法は慎重に使用する必要があります。同じFSIDでデスティネーションSVMに書き込みを試みる前に、ソースSVMが停止していることを確認する必要があります。

NFSv3のセキュリティに関する考慮事項

NFSv3はNFSv4.xよりも安全性が低いとみなされますが、だからといってセキュリティ対策が講じられていないわけではありません。以降のセクションでは、ONTAPに該当するNFSv3のセキュリティに関する考慮事項について説明します。

NFSv3エクスポートルール

NFSマウントを保護する方法の1つは、エクスポートルールを使用する方法です。ストレージ管理者は、エクスポートルールを使用して、ボリュームまたはqtreeへのアクセスを環境内の特定のクライアントに対して遮断することができます。たとえば、アクセスを許可するエクスポートルールにNFSクライアントが含まれていない場合、そのクライアントはエクスポートをマウントできません。

エクスポートルールは、特定のクライアントに読み取り、書き込み、setuid権限、およびファイル所有権を変更するためのアクセスを許可するかどうかを制限する方法も提供します。ルールでは、マウントを許可する特定のNFSバージョンや、許可するセキュリティの種類（AUTH_SYS、Kerberosなど）を定義できます。

エクスポートルールのセキュリティ制限

エクスポートルールは、NFSマウントを保護する1つの方法にすぎませんが、企業の標準に近づくほどのセキュリティは提供されません。エクスポートルールは、clientmatchエントリに基づいてアクセスを制限します。つまり、エクスポートルールでKerberos認証などの他のセキュリティ概念も要求されていないかぎり、clientmatchリスト内のIPアドレスがスプーフィングされ、他の認証要件なしでエクスポートにアクセスできる可能性があります。

NFSv3権限

NFSv3には、エンドユーザへのアクセスを制御するための基本的なファイルおよびフォルダ権限が用意されています。これらの権限は、RFC 1813（22ページ以降）のNFSモードビットの定義に従っています。

NFSv3権限のセキュリティ制限

モードビット権限は、所有者、グループ、その他のすべてのユーザに適用されます。つまり、基本的なNFSv3では、ユーザアクセスをきめ細かく制御することはできません。ONTAPはPOSIX ACLをサポートしていないため、詳細なACLはNFSv3の次のシナリオでのみ実行できます。

- 有効なUNIXからWindowsへのユーザマッピングが設定されたNTFSセキュリティ形式のボリューム（CIFSサーバが必要）。
- NFSv4.x ACLを適用するためにNFSv4.xをマウントする管理クライアントを使用して適用されるNFSv4.x ACL。

また、モードビットでは、NTFSまたはNFSv4.x ACLと同等の権限レベルは提供されません。次の表に、NFSv3モードビットとNFSv4.x ACLの権限の比較を示します。NFSv4.x ACLの詳細については、https://linux.die.net/man/5/nfs4_aclを参照してください。

- 表6 に、NFSv3モードビットとNFSv4.x ACLの精度の比較を示します。

表6）NFSv3モードビットとNFSv4.x ACLの精度

NFSv3モード ビット	NFSv4.x ACL
• 実行時にユーザIDを設定 • 実行時にグループIDを設定 • スワップされたテキストを保存（POSIXでは定義されていません） • 所有者の読み取り権限 • 所有者の書き込み権限 • ファイルの所有者の実行権限、またはディレクトリの所有者の検索（検索）権限 • グループの読み取り権限 • グループの書き込み権限 • ファイルに対するグループの実行権限、またはディレクトリ内のグループの検索（検索）権限 • 他のユーザーの読み取り権限 • 他のユーザーの書き込み権限 • ファイルに対する他のユーザーの実行権限、またはディレクトリ内の他のユーザーの検索（検索）権限	• ACEタイプ（許可/拒否/監査） • 継承フラグ： – ディレクトリ継承 – ファイル継承 – no-propagate-inherit – 継承のみ • 権限： – read-data（ファイル）/list-directory（ディレクトリ） – write-data（ファイル）/create-file（ディレクトリ） – append-data（ファイル）/create-subdirectory（ディレクトリ） – execute（ファイル）/change-directory（ディレクトリ） – 削除 – delete-child – 読み取り属性 – 書き込み属性 – read-named-attributes – write-named-attributes – 読み取りACL – 書き込みACL – write-owner • 同期

最後に、NFSグループメンバーシップ（NFSv3とNFSv4.xの両方）は、RPCパケットの制限に従って最大16に制限されます。ONTAPには、「補助GID–NFSの16 GID制限への対処」に記載されているこれらの制限を超えるオプションが用意されています。

NFSv3のユーザIDとグループID

NFSv3のユーザIDとグループIDは、名前ではなく数値IDとして認識されます。UNIXセキュリティ形式のボリュームおよびqtreeにNFSv4.x ACLが設定されていない場合、ONTAPはこれらの数値IDの名前解決を行いません。NTFSセキュリティ形式およびNFSv4.x ACLを使用する場合、ONTAPは数値IDを有効なWindowsユーザに解決してアクセスをネゴシエートします。詳細については、「NFSクレデンシャルの表示と管理」を参照してください。

NFSv3ユーザIDとグループIDのセキュリティ制限

クライアントとサーバは、数値IDで読み取りまたは書き込みを試行しているユーザが実際に有効なユーザであることを確認する必要はありません。暗黙的に信頼されているだけです。これにより、既知の数値IDをスプーフィングするだけで、ファイルシステムが潜在的な違反に見舞われる可能性があります。NFSv3でのセキュリティホールの発生を防ぐために、NFSでKerberosを実装すると、ユーザはユーザ名とパスワードまたはkeytabファイルで認証され、マウントへのアクセスを許可するKerberosチケットを取得する必要があります。これをNFSv4.xまたはNTFS ACLと組み合わせて使用すると、NFSv3の数値IDによるパフォーマンスリスクの軽減に役立ちます。

NFSv3セキュリティの高度な概念

次のセクションでは、NFSv3の2つの高度なセキュリティの概念について説明します。Advancedという用語は、非標準的な概念、または複数の設定手順を含む概念を指します。

NFSv3でのACLの使用

前述したように、ONTAPのNFSv3ではPOSIX ACLはサポートされていません。ただし、ACLを使用してNFSv3のユーザおよびグループのアクセスを制御する方法は2つあります。

- **NTFS ACL**：ONTAPを使用すると、同じボリューム内のデータセットを複数のNASプロトコルに同時に提示できます。たとえば、ボリュームはNFSおよびSMB経由でクライアントをホストできますが、ロックや権限などのプロトコル固有の機能は、データの耐障害性とセキュリティを確保するためにONTAPによってネゴシエートされます。

  CIFS / SMBアクセスとNFSアクセスの両方にONTAPを使用する場合は、Windowsクライアントで設定したファイルおよびフォルダの権限を使用してNFSv3クライアントアクセスを管理できます。CIFS / SMBサーバを作成し、ボリュームのセキュリティ形式でNTFS権限セマンティクスを使用している場合、NTFS ACLを含むボリュームへのNFSアクセスでONTAPが必要とするUNIXからWindowsへのユーザ名マッピングによって、NFSv3クライアントはNTFS ACLに準拠します。この機能の詳細については、「マルチプロトコルNAS」を参照してください。

- **NFSv4.x ACL**：NFSv4.xはONTAPとNFSv3でサポートされ、NTFS ACLと同じようにきめ細かなファイル権限とフォルダ権限を提供します。ONTAPでは、NFSv3クライアントが優先する管理者クライアントを使用してNFSv4.x ACLを設定することもできます。NFSv3クライアントがNFSv4.x ACLを含むマウントにアクセスしようとすると、ONTAPは数値のユーザIDをNFSv4.xのネームマッピング要件に従って有効なUNIXユーザに解決し、NFSv4.x ACLが適切に維持されるようにします。NFSv4.x ACLの詳細については、「NFSv4.x ACL」を参照してください。

NFSv3でのNFS Kerberosの使用

KerberosはNFSv3でも使用できますが、次の点に注意してください。

- NFSv3は、NFSプロトコルに加えて、いくつかの補助プロトコルで構成されています。NFSv3でKerberosを使用する場合は、NFSパケットのみがKerberosを使用します。mount、portmapなどはKerberosを使用しません。
- NFSv3とKerberosのエクスポートポリシールールでは、ONTAP 8.2P5以前のバージョンを使用している場合に補助プロトコルでKerberosがサポートされないことを反映するために、設定にsysとkrb5 *の両方を使用する必要があります。最新のリリースでは、エクスポートポリシールールをこのように変更する必要はありません。詳細については、バグ756081を参照してください。

  注：NFSでのKerberosの使用の詳細については、TR-4616：『NFS Kerberos in ONTAP』を参照してください。

ONTAPファイアウォールポリシーによるポートマップのブロック

ONTAP 9.3以前では、サードパーティ製のファイアウォールではなく組み込みのONTAPファイアウォールを使用するネットワーク構成では、ポート111でportmapサービス（rpcbind）に常にアクセスできるため、潜在的なセキュリティの脆弱性が発生しました。ONTAP 9.4以降では、ファイアウォールポリシーを変更して、portmapサービスへのアクセスを許可するかどうかをLIFごとに制御できます。新しいファイアウォールポリシーはmgmt-nfsで、デフォルトで次のルールが適用されています。

```
        mgmt-nfs
                    dns         0.0.0.0/0, ::/0
                    http        0.0.0.0/0, ::/0
                    ndmp        0.0.0.0/0, ::/0
                    ndmps       0.0.0.0/0, ::/0
                    ntp         0.0.0.0/0, ::/0
                    portmap     0.0.0.0/0, ::/0
                    snmp        0.0.0.0/0, ::/0
```

考慮事項：

- アップグレード時に、ONTAPは既存のすべてのファイアウォールポリシー（デフォルトまたはカスタム）にportmapサービスを追加します。
- 新しいクラスタやIPspaceを作成した場合、portmapサービスはデフォルトのデータ ポリシーにのみ追加され、デフォルトの管理ポリシーまたはクラスタ間ポリシーには追加されません。
- 必要に応じて、デフォルトまたはカスタムのポリシーにportmapサービスを追加したり削除したりできます。

親からのグループ所有者の継承

一部の競合システムから移行する場合、新しいファイルおよびディレクトリを作成するときに親ディレクトリのグループ所有者を継承するオプションが使用されることがあります。ONTAPにはこの特定のオプションはありませんが、権限にスティッキビットフラグを使用することで同様の機能を実現できます。

1. スティッキビットを設定するには、次のコマンドを実行します。

```
# chmod 2775 setguid/
```

2. 親ディレクトリの所有者を変更します。

```
# chown prof1:ProfGroup setguid/
```

これで、フォルダのグループモードビットの実行部分に「s」が表示されます。

```
# ls -la | grep setguid
drwxrwsr-x   2 prof1            ProfGroup                  4096 Oct 6 16:10 setguid
```

新しいファイルが作成されると、親ディレクトリからグループが継承されます。この場合、rootはユーザです。

```
# cd setguid/
# id
uid=0(root) gid=0(root) groups=0(root)
# touch newfile
# ls -la
total 8
drwxrwsr-x 2 prof1 ProfGroup 4096 Oct 6 2020 .
drwxrwxrwx 20 root root       4096 Oct 6 16:10 ..
-rw-r--r--  1 root ProfGroup     0 Oct 6 2020 newfile
```

NFSv4.xに関する考慮事項

次のセクションでは、ONTAPのNFSv4.xの機能、既知の問題、および考慮事項について説明します。

NFSv4.xの有効化

ご使用の環境でNFSv4.xとONTAPの使用を開始するための手順を次に示します。

1. で、NFS IDのドメイン文字列を `/etc/idmapd.conf -v4-id-domain` NFS SVMのオプションと同じ値に設定します。
2. NFSv4.xクライアントにアクセスするユーザとグループがONTAP SVMにも存在している（またはそこから照会できる）ことを確認します。これらのユーザとグループは、名前と大文字と小文字を区別する必要があります。
   - たとえば、NFSクライアントのjohn@DOMAIN.COMは、 ONTAP NFSサーバのjohn@DOMAIN.COMと一致する必要があります。
3. UNIX IDにLDAPまたはNISを使用する場合は、ユーザとグループの検索で想定されるIDとグループメンバーが返されることを確認します。
4. ボリュームのエクスポートポリシールールは、NFSv4をプロトコルとして許可するように設定する必要があります。
5. 環境内のデータLIFで、許可するプロトコルとしてNFSを使用する必要があります（net int show -fields allowed-protocols）。
6. 必要なバージョンのNFSv4.xを有効にする必要があります。バージョン4.1のみが必要な場合は、そのバージョンのみを有効にし、バージョン4.0を無効にします。

7. NFSv4.x ACLが必要な場合は、NFSv4.x ACLを使用する特定のバージョンのNFSv4.x ACLを有効にする必要があります。（`-v4.0-acl`、 `-v4.1-acl`）
8. クライアントは、NFSサーバがサポートする最も高いNFSバージョンをネゴシエートします。一部のクライアントでNFSv3が必要な場合は、マウント方法を変更する必要があります。

NFSv4.xを使用する利点

環境でNFSv4.xを使用する利点は次のとおりです。

- ファイアウォールとの親和性。NFSv4では1つのポート（2049）しか使用しません。
- 高度でアグレッシブなキャッシュ管理（NFSv4.xの委譲など）
- 強固なRPCセキュリティ種別。暗号化を実装します。
- 国際化。
- 複合操作。
- TCPでのみ動作。
- ステートフル プロトコル（NFSv3はステートレス）。
- 効率的な認証メカニズムのためのKerberos設定：
  - clustered ONTAP 8.2.x以前でのDESと3DESの暗号化のサポート
  - 8.3以降でのAESのサポート
- リファーラルを使用した移行（dNFS）。
- UNIXおよびWindowsと互換性のあるアクセス制御のサポート。
- 文字列ベースのユーザ識別子とグループ識別子。
- pNFSを介したデータへの並行アクセス（NFSv4.0は該当せず）

それぞれのユースケースを別々に扱うことが重要です。NFSv4.xはすべてのワークロード タイプに適しているわけではありません。NFSv4.xを環境全体に展開する前に、必要な機能とパフォーマンスについてテストするようにしてください。

注： ONTAPは現在、NFSv4.xセッショントランキングをサポートしていません。

NFSv4.x処理のパフォーマンス強化

NetAppは、ONTAPリリースごとにパフォーマンスの向上に絶えず努めています。NFSでは今後もNFSv4.xのパフォーマンスが最優先事項となっています。以下に、導入されたONTAPリリースとともにパフォーマンスの強化点を示します。NFSのパフォーマンスを最大限に高めるには、パッチが適用された最新のONTAPリリースを常に実行してください。

NFSv4.xファストパス（ONTAP 8.2で導入）

ONTAP 8.2以降では、NFSv4の読み取りと書き込みのパフォーマンスを向上させるためにNFSファストパスが導入されました。これは、ボリュームをホストしているノードに対してローカルなLIFでデータ要求が行われたときに、NFSv4パケットからONTAP主体のパケットへの内部処理をバイパスすることで実現されました。pNFSやリファーラルなどの他の機能と組み合わせることで、読み取りおよび書き込み要求ごとにローカライズされたデータが保証されるため、一貫してNFSv4ファストパスを使用できます。NFSファストパスはNFSv3では常に使用されていました。NFSファストパスはデフォルトで有効になります。

NFSv4.xのマルチスレッド処理（ONTAP 8.2で導入）

ONTAP 8.2以降では、NFSv4.xの読み取り処理と書き込み処理に対してマルチプロセッサがサポートされるようになりました。ただし、メタデータ処理は引き続きシングル スレッドで行われます。以前のリリースでは、NFSv4.xの読み取りおよび書き込み処理はシングル スレッドだったため、プロトコル ドメインのCPUがボトルネックとなることがありました。大量の読み取りと書き込みが行われるNFSv4.xワークロードに複数のCPUを使用するNetApp システムでは、読み取りおよび書き込み処理に複数のプロセッサを使用することでスループットが大幅に向上します。

注：NFSv3では、読み取りと書き込みに常に複数のプロセッサが使用されています。メタデータ処理にも複数のプロセッサが使用されます。

NFSv4.x–ストリーミングワークロードタイプのパフォーマンスの向上（ONTAP 9.0で導入）

ONTAP 9.0では、大規模なI/Oのサポートが追加され、VMware、Oracle、SAP HANAなどのストリーミングワークロードのNFSv4.1でのパフォーマンスが向上しました。これにより、NFS（v3と4.xの両方）で読み取りと書き込みの両方に最大1MBを使用できるようになりました。

NFSv4.x–メタデータワークロードのパフォーマンスの向上（ONTAP 9.5で導入）

ONTAP 9.5では、メタデータワークロードを改善するために、次のような多くの改善が追加されています。

- NFSv4.0のキャッシュI/Oのサポート
- NFSv4.xメタデータ処理の最適化
- キャッシュの向上
- ロック性能の向上
- StorePoolの上限の引き上げ

NFSv4.x FlexGroupボリュームのサポート（ONTAP 9.7で導入）

ONTAP 9.5で追加されたメタデータパフォーマンスの向上に加えて、ファイル取り込み処理を並列化することで、FlexGroupボリュームのサポートによってメタデータワークロードのパフォーマンスも向上します。FlexGroupボリュームの詳細については、TR-4571：『NetApp FlexGroup Volumes Best Practices and Implementation』を参照してください。

メタデータ処理のNFSv4.xパフォーマンスの強化（ONTAP 9.8）

ONTAP 9.8では、メタデータ比率の高いワークロードを処理する際のNFSv4.xの全体的なパフォーマンスを向上させるために、いくつかの機能拡張が追加されました（ソフトウェアビルドに関する標準的なNASベンチマークテストなど）。その結果、これらのワークロードは大幅に改善され、1ミリ秒のレイテンシで達成されたピークIOPSはNFSv3のパフォーマンスに少し近付きました。

これには、次のものが含まれます。

- 書き込みロック使用率の向上
- QoSパフォーマンスの最適化
- リプレイ処理の改善（バグ1281571を参照）
- 化合物操作の統合
- Storepoolの最適化
- オープン/クローズの機能拡張
- nconnectのサポート（詳細については、を参照してくださいnconnect）
- リンク解除操作の並列化
- READDIR操作の並列化

ONTAP 9.9.1でのNFSv4.xパフォーマンスの最適化

ONTAP 9.9.1ではさらに最適化が追加され、メタデータ比率の高いワークロードを処理する際のNFSv4.xの全体的なパフォーマンスが向上しました（ソフトウェアビルドに関する標準的なNASベンチマークテストなど）。その結果、これらのワークロードはONTAP 9.8よりも大幅に改善され、NFSv3の約22%の範囲内で1ミリ秒のレイテンシでピークIOPSを達成しました。

- ルックアップおよびクローズ操作時の属性のプリフェッチ
- オープン時のアクセス呼び出しのプリフェッチによるメタデータ処理の削減
- 書き込み最適化後の属性の取得

- プリフェッチされた属性を保存するメモリプール
- FlexGroupの最適化（FlexGroupボリュームの詳細をキャッシュして検索を最適化）
- エクスポートチェックのパス長の削減

NFSv4.0

NetApp ONTAP NFSv4.xの実装には次の機能があります。

- **書き込み順序。**データ ブロックを、データ バッファ内の順序で共有ストレージに書き込むことができます。
- **同期書き込みの永続性：**ONTAP（clustered Data ONTAPと7-Mode）では、同期書き込み呼び出しから戻った時点で、すべてのデータが永続的なストレージに確実に書き込まれます。
- **分散ファイルロック：**2つのサーバに同時にロックを割り当てることなく、共有ストレージに対して排他的なロックを要求および取得することができます。
- **一意の書き込み所有権。**ONTAP（clustered Data ONTAPと7-Mode）では、ファイルロックがファイルへの書き込みが可能な唯一のサーバプロセスであることが保証されます。ONTAPがロックを別のサーバに転送したあと、前の所有者がキューに登録した保留中の書き込みは失敗します。

NFSv3からNFSv4.xへの移行：考慮事項

次のセクションでは、NFSv3からNFSv4.xに移行する際に対応する必要がある考慮事項について説明します。NFSv3からNFSv4.xに移行する場合、NFSv4.xをオンにするだけではNFSv4.xは予期したとおりに機能しません。次のような項目に対応する必要があります。

- ドメイン文字列 / IDのマッピング
- ストレージ フェイルオーバーに関する考慮事項
- ネーム サービス
- ファイアウォールに関する考慮事項
- エクスポート ポリシー ルールに関する考慮事項
- クライアントのサポート
- NFSv4.xの機能

NFSv4.xプロトコルの詳細（NFSv4.2に関する情報を含む）については、SNIA overview of NFSv4を参照してください。

NFSv4.x IDドメインマッピング

既存の設定とインフラをNFSv3からNFSv4に移行する際には、移行前に一部の環境で変更が必要になります。その1つがIDドメインマッピングです。

NFSv3クライアントがマウントにアクセスすると、数字のIDがNFSサーバに渡されます。アクセスのすべての部分でNFSv3のセマンティクスに対応したUNIXセキュリティが使用されていれば、それ以降のID検索は必要ありません。

NFSv4.xクライアントがマウントにアクセスすると、クライアントからユーザまたはグループプリンシパルを含むサーバ（name@domain.com）に名前文字列が渡されます。

その後、サーバは、ネームサービスとNFSサーバの設定を通じて、同じ名前（大文字と小文字が区別されます）を持つプリンシパルについて認識しているかどうかを確認しようとします。この名前文字列がクライアントとサーバの両方に存在しない場合は、クライアントのNFSv4.x構成ファイルで定義されているnobodyユーザにユーザが引き下げられます。これにより、NFSv3に比べてセキュリティが強化されます。

名前文字列のバイパス：数値ID

場合によっては、セキュリティのためではなく、ロックメカニズムのためにNFSv4.xを使用したいと考えるストレージ管理者もいます。または、クライアント/サーバ名文字列の設定オーバーヘッドに対処したくない場合もあります。

その場合、と呼ばれるオプションがあります `v4-numeric-ids`。このオプションを有効にすると、クライアントがネームマッピングにアクセスできない場合に、ユーザ名フィールドとグループ名フィールドで数値IDを送信できます。数値IDを受け取ったサーバは、対応する数値を含むv2 / v3のUIDまたはGIDで表されるユーザと同じユーザを表すものとして処理します。クライアントに一致する名前文字列がある場合、クライアントはNumericIDではなく名前文字列を使用します。クライアントとサーバのユーザ名が一致していてもドメイン文字列が一致していない場合は、数値は使用されず、ユーザ名/グループ名はに戻ります `nobody`。これはrootユーザの一般的なシナリオです。rootユーザは常にクライアントとサーバに存在し、ONTAPのNFSv4のID文字列はデフォルトでになっているためです。`defaultv4iddomain.com`NFSクライアントでは `idmapd.conf` 、ファイル内のドメイン文字列設定がデフォルトではない（NFSv4ドメインのDNSドメインにフォールバックする）ため、このシナリオでは多くの場合、不一致が発生します。

基本的に、このオプションを選択すると、NFSv4.xはNFSv3のように動作します。このオプションのデフォルト値はenabledです。拡張グループのサポートに関する考慮事項については、「数値ID認証（NFSv3およびNFSv4.x）に関する考慮事項」を参照してください。

ストレージ フェイルオーバーに関する考慮事項

NFSv4.xで使用されるロック モデルは、NFSv3とまったく異なります。NFSv4.xのロックはリースベースのモデルで、NFSv3（NLM）と違ってプロトコルに統合されています。

ONTAPのドキュメントから：

```
In accordance with RFC 3530, ONTAP "defines a single lease period for all state held by an NFS
client. If the client does not renew its lease within the defined period, all states associated
with the client's lease may be released by the server." The client can renew its lease explicitly
or implicitly by performing an operation, such as reading a file.
Furthermore, ONTAP defines a grace period, which is a period of special processing in which
clients attempt to reclaim their locking state during a server recovery.
```

表7）NFSv4.xのロック関連用語

期間	定義（RFC 3530に準拠）
リース期間	ONTAPがクライアントに解除不能なロックを付与する期間
猶予期間	サーバリカバリ中にクライアントが自身のロック状態をONTAPに再要求する期間。
ロック	他に特に記載がないかぎり、レコード（バイト範囲）ロックとファイル（共有）ロックの両方を表します。

NFSv4.xロックの詳細については、本ドキュメントの「NFSv4ロック」に関するセクションを参照してください。 この新しいロック方法とステートフルプロトコルのため、NFSv4.xプロトコルのストレージフェイルオーバーの動作はNFSv3とは異なります。詳細については、本ドキュメントの「NFSによるノンストップオペレーション」を参照してください。

ネーム サービス

NFSv4.xの使用を決定する際には、NetAppのベストプラクティスとして、NFSv4.xユーザをLDAPやNISなどのネームサービスで一元化することを推奨します。これにより、すべてのクライアントとONTAP NFSサーバが同じリソースを利用できるようになり、環境全体ですべての名前、UID、およびGIDの一貫性が保証されます。ネームサービスの詳細については、TR-4835：『How to Configure LDAP in ONTAP』およびTR-4668：『Name Services Best Practices』を参照してください。

ファイアウォールに関する考慮事項

NFSv3では、ポート2049に加えて、NLMやNSMなどの補助プロトコル用にいくつかのポートを開く必要がありました。NFSv4.xで必要なポートは2049だけです。同じ環境でNFSv3とNFSv4.xを使用する場合は、関連するすべてのNFSポートを開きます。これらのポートについては、「ONTAPのデフォルトのNFSポート」を参照してください。 ファイアウォールの詳細とガイダンスについては、「NFSセキュリティのベストプラクティス」を参照してください。

ボリュームの言語に関する考慮事項

ONTAPでは、ボリュームに特定の言語を設定できます。この機能は、英語にはない文字を使用する言語（日本語、中国語、ドイツ語など）でのファイル名の多言語対応を目的としています。RFC 3530では、NFSv4.xを使用する場合にUTF-8が推奨されると記載されています。

```
11. Internationalization

   The primary issue in which NFS version 4 needs to deal with
   internationalization, or I18N, is with respect to file names and
   other strings as used within the protocol. The choice of string
   representation must allow reasonable name/string access to clients
   which use various languages. The UTF-8 encoding of the UCS as
   defined by [ISO10646] allows for this type of access and follows the
   policy described in "IETF Policy on Character Sets and Languages",
   [RFC2277].
```

ボリュームの言語を変更する場合は、変更後にボリューム内のすべてのファイルにアクセスして、言語の変更がすべて反映されていることを確認する必要があります。`ls -lR` ファイルの再帰的なリストにアクセスするには、simpleを使用します。ファイル数が多い環境の場合は、XCPを使用してファイルをすばやくスキャンすることを検討してください。

エクスポート ポリシー ルール

NFSv3用に設定された環境で、エクスポートポリシールールオプション `-protocol` がNFSv3のみを許可するように制限されている場合は、NFSv4を許可するようにオプションを変更する必要があります。さらに、NFSv4.xクライアントのアクセスのみを許可するようにポリシー ルールを設定することもできます。

例：

```
cluster::> export-policy rule modify -policy default -vserver NAS -protocol nfs4
```

詳細については、ご使用のバージョンのONTAPの製品ドキュメントを参照してください。

クライアントに関する考慮事項

NFSv4.xを使用する場合は、NFSサーバ同様にクライアントについても考慮することが重要です。NFSv4.xの設定に関する具体的な質問については、オペレーティングシステムのベンダーにお問い合わせください。

NFSv4.xを実装する際のクライアントに関する考慮事項は次のとおりです。

注： その他の考慮事項が必要になる場合があります。

- NFSv4.xがサポートされている。
- fstabファイルとNFS構成ファイルが正しく設定されている。マウント時に、クライアントは使用可能な最も高いNFSバージョンをNFSサーバとネゴシエートします。クライアントでNFSv4.xが許可されていないか、fstabでNFSv3が指定されている場合は、マウント時にNFSv4.xが使用されません。
- idmapd.confファイルに正しいNFSv4.x IDドメインなどの適切な設定が適用されている。
- クライアントがローカルのpasswdファイルとgroupファイルに同一のユーザ/グループおよびUID / GID（大文字と小文字の区別を含む）を含んでいるか、NFSサーバ/ ONTAP SVMと同じネームサービスサーバを使用している。
- クライアントでネームサービスを使用する場合は、クライアントがネームサービス用に適切に設定されていること（nsswitch.conf、ldap.conf、sssd.confなど）と、適切なサービスが開始、実行され、ブート時に開始されるように設定されていることを確認します。

- NFSv4.xサービスが開始されて実行中であり、さらにブート時に開始されるように設定されている。

NFSv4.xをNFSv3と同等に機能させる

場合によっては、NFSv4.xの選択がセキュリティ上の懸念からではなく、アプリケーションベンダーやサービスプロバイダがNFSv4.xを使用する必要があるために行われることもあります。NFSv4.x IDの有効なドメイン情報を提供し、ユーザを適切にマッピングできるインフラがないと、状況が複雑になる可能性があります。NFSv4.xをNFSv3のように機能させる場合は、次の処理を実行します。

- `-v4-numeric-ids` NFS SVMでが有効になっていることを確認する（デフォルトは有効）
- NFSv4.x IDドメインがSVM（`-v4-id-domain`）およびクライアント `idmapd.conf` のファイル（またはクライアントのDNSドメイン名と同じ）で一致することを確認します。
- `/sys/module/nfsd/parameters/nfs4_disable_idmapping` NFSクライアントで（または同等の）をYに設定します。

これらの手順を実行すると、クライアントはIDマッピングをバイパスし、サーバとクライアントの両方にユーザ名が存在しない場合は数値IDにフォールバックします。

NFSv4の特徴と機能

NFSv4.xはNFSプロトコルの進化形であり、NFSv3にリファーラル、委譲、pNFSなどの新機能が追加されて強化されています。これらの機能については本ドキュメントで説明しています。NFSv4.xの実装に関する設計上の決定には、これらの機能を考慮する必要があります。

NFSv4ユーザIDマッピング

このドキュメントで前述したように（「NFSv4.x IDドメインのマッピング」セクションで）、NFSv4.xのクライアントおよびサーバは、セキュリティを強化するためにユーザIDドメイン文字列のマッピングを試みます。数値IDを使用する場合、ONTAPには（`-v4-numeric-ids`）名前文字列の要件を回避するためのNFSオプションがあります。

NFSv4.x ACL

NFSv4.xプロトコルでは、ACLの形式でアクセス制御を提供できます。ACLは、CIFSのACLと概念的には似ています。NFSv4 ACLは個々のアクセス制御エントリ（ACE）で構成され、各ACEがサーバへのアクセス制御ディレクティブを提供します。ONTAPのデフォルトのACEは400で、設定可能なNFSオプションで最大1、024個のACEをサポートします（`-v4-acl-max-aces`）。

NFSv4 ACLを有効化する利点

以下に、その利点を示します。

- ファイルやディレクトリへのユーザ アクセスの詳細な制御
- NFSセキュリティの向上
- CIFSとの相互運用性の向上
- AUTH_SYSセキュリティでのユーザあたりの最大NFSグループ数（16）の解除
  - ACLではGID解決の必要がないため、実質的にGIDの制限が排除されます。

NFSv4 ACLとSMBクライアントの互換性

NFSv4 ACLはWindowsのファイルレベルのACL（NTFS ACL）とは異なりますが、機能は似ています。ただし、マルチプロトコルNAS環境では、NFSv4.x ACLが `ntacl-display- permissive-perms is-unix-nt-acl-enabled` 設定されている場合、SMB2.0以降を使用するクライアントは、NFSオプションと. CIFS/SMBオプションが設定されていても、WindowsのセキュリティタブからACLを表示できません。詳細については、バグ928026を参照してください。

NFSv4 ACLの仕組み

クライアントがSETATTR操作でファイルにNFSv4 ACLを設定すると、NetApp ストレージ システムは既存のACLに替わってそのACLをオブジェクトに設定します。ファイルにACLが設定されていない場合、ファイルのモード権限はOWNER@、GROUP@、およびEVERYONE@から計算されます。ファイルにSUID / SGID / STICKYのいずれかのビットが設定されている場合、それらのビットは影響を受けません。

クライアントがGETATTR操作でファイルのNFSv4 ACLを取得すると、NetApp システムはオブジェクトに関連付けられたNFSV4 ACLを読み取り、ACEのリストを作成してクライアントに返します。ファイルにNT ACLまたはモード ビットが設定されている場合は、モード ビットからACLが作成されてクライアントに返されます。

ACLにDENY ACEが存在する場合はアクセスが拒否され、ALLOW ACEが存在する場合はアクセスが許可されます。ただし、ACLにどちらのACEも存在しない場合も、アクセスが拒否されます。

セキュリティ記述子は、セキュリティACL（SACL）と随意ACL（DACL）で構成されます。NFSv4がCIFSと連動する場合は、DACLはNFSv4とCIFSに1対1でマッピングされます。DACLは、ALLOW ACEとDENY ACEで構成されます。

NFSv4.x ACLが設定されたファイルまたはフォルダに対して基本的なchmodを実行すると、NFSオプションが `v4-acl-preserve` 有効になっていないかぎりACLが削除されます。

NFSv4 ACLを使用しているクライアントは、システム上のファイルとディレクトリにACLを設定し、そのACLを表示することができます。ACLが設定されたディレクトリに新しいファイルまたはサブディレクトリを作成すると、新しいファイルまたはサブディレクトリには、該当する継承フラグが設定されたACL内のACEがすべて継承されます。アクセス チェックでは、CIFSユーザがUNIXユーザにマッピングされ、マッピングされたUNIXユーザとそのユーザのグループ メンバーシップがACLに照らしてチェックされます。

ファイルやディレクトリにACLが設定されている場合は、ファイルやディレクトリのアクセスに使用されるプロトコルの種類（NFSv3、NFSv4、またはCIFS）にかかわらず、そのACLを使用してアクセスが制御されます。このACLは、システムでNFSv4が有効でなくなったあとも使用されます。

親ディレクトリのNFSv4 ACLのACEに正しい継承フラグが設定されていれば、ファイルやディレクトリは該当するACEを継承します（必要な変更が加えられる可能性があります）。

ファイルやディレクトリがNFSv4要求によって作成される場合、作成されるファイルやディレクトリのACLは、ファイル作成要求にACLが含まれているか、または標準のUNIXファイル アクセス権限のみが含まれているかによって異なります。また、親ディレクトリにACLが設定されているかどうかによっても異なります。

- 要求にACLが含まれる場合は、そのACLが使用されます。
- 要求に標準のUNIXファイル アクセス権限のみが含まれ、親ディレクトリにACLがない場合は、クライアントのファイル モードを使用して標準のUNIXファイル アクセス権限が設定されます。
- 要求に標準のUNIXファイルアクセス権限のみが含まれ、親ディレクトリに継承できないACLがある場合は、要求で渡されたモードビットに基づいてデフォルトのACLが設定されます。
- 要求に標準のUNIXファイル アクセス権限のみが含まれ、親ディレクトリにACLがある場合、親ディレクトリのACLのACEに該当する継承フラグが設定されていれば、それらのACEが新しいファイルやディレクトリに継承されます。

  注： 親ACLは `-v4.0-acl` 、がに設定されている場合でも継承され `off`ます。

umaskおよびACLの継承でのNFSv4 ACLの動作

NFSv4 ACLでは、ACLを継承できます。ACLの継承とは、NFSv4 ACLが設定されたオブジェクトの下に作成されたファイルやフォルダで、ACL 継承フラグの設定に基づいてACLを継承できることを意味します。

umask は、ディレクトリ内にファイルやフォルダを作成する際の権限レベルを制御するために使用します。詳細については、を参照してくださいumask。

デフォルトでは、ONTAPは継承されたACLをumaskによって上書きすることを許可しています。これはRFC 5661に従った動作です。umaskを使用してACL継承の動作を調整するには、オプションを有効にし `-v4-inherited- acl-preserve`ます。

ACLノケイシキ

NFSv4.x ACLには特定の形式があります。次に、ファイルに設定されたACEの例を示します。

```
A::ldapuser@domain.netapp.com:rwatTnNcCy
```

上記の例では、のACL形式のガイドラインに従っています。

```
type:flags:principal:permissions
```

Aのタイプは許可することを意味します。この例ではフラグ（flags）が設定されていません。これは、プリンシパル（principal）がグループではなく、継承を含んでいないためです。また、ACEは監査エントリではないため、監査フラグを設定する必要もありません。NFSv4.x ACLの詳細については、http://linux.die.net/man/5/nfs4_aclを参照してください。

NFSv4.x ACLが適切に設定されていないと、ACLが想定どおりに動作しなかったり、ACLの変更が適用されずにエラーが発生したりすることがあります。

エラーの例は次のとおりです。

```
Failed setxattr operation: Invalid argument
Scanning ACE string 'A::user@rwaDxtTnNcCy' failed.
```

明示的なDENY

NFSv4権限には、OWNER、GROUP、EVERYONEの明示的なDENY属性を含めることができこれは、NFSv4 ACLがdefault-denyであるためです。つまり、ACEによってACLが明示的に許可されていない場合、ACLは拒否されます。明示的なDENY属性は、明示的かどうかに関係なく、すべてのアクセスACEを上書きします。

Deny ACEは、Dの属性タグで設定されます。

例：

```
sh-4.1$ nfs4_getfacl /mixed
A::ldapuser@domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A:g:GROUP@:rxtncy
D:g:GROUP@:waDTC
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC
```

DENY ACEは混乱し複雑になる可能性があるため、可能なかぎり使用しないでください。DENY ACEを設定すると、アクセスを許可されるはずのユーザがアクセスを拒否される場合があります。これは、NFSv4 ACLの順序がその評価に影響するためです。

上記のACEのセットは、モードビットの755に相当します。つまり、次のことを意味します。

- 所有者にはフル アクセス権がある。
- グループには読み取り専用アクセス権がある。
- それ以外には読み取り専用アクセス権がある。

ただし、775と等しくなるように権限が調整されていても、EVERYONEに明示的なDENYが設定されているとアクセスが拒否される可能性があります。

明示的な拒否の例については、「NFSv4.x ACLの明示的な拒否の例」を参照してください。

NFSv4 ACL保持

デフォルトでは、NFSv4 ACLはファイルまたはフォルダに設定されたモード ビットの影響を受けます。NFSv4 ACEが設定されている場合にを chmod 使用すると、ACEは削除されます。これは v4-acl- preserve オプションで制御されます。例については、本ドキュメントの「NFSv4.x ACL保持の例」を参照してください。

注： デフォルトでは、このオプションは[有効]に設定されています。NetAppでは、特定のユースケースが発生しない限り、このオプションを変更しないことを推奨しています。

NFSv4 ACLを使用してWindowsで作成されたファイルからのモードビット表示の動作

マルチプロトコルNAS環境（SMBとNFSの両方を使用して同じデータにアクセス）では、NFSv4 ACLによって望ましくない動作が発生する可能性があります。このような場合、ONTAP 9.7以前のバージョンでは、NFSクライアントでSMBを介して作成されたファイルでNFSv4 ACLを使用している場合にが表示されます。ONTAP 9.8以降では is-inherit-modebits-with-nfsv4acl-enabled、この問題を解決するためのCIFSサーバオプションが導入されています（デフォルトは無効）。モードビットを適切に表示するには、このオプションをenabledに設定します。

詳細については、バグ820848を参照してください。

NFSv4 ACLが適用されているファイルおよびフォルダでのchmodの禁止

場合によっては chmod 、NFSv4 ACLが設定されているファイルやフォルダに対して、ファイル所有者からであってもコマンドが機能しないようにすることができます。これは -restrict-chmod- acl、ONTAP 9.8のオプション（diag権限）で制御されます。デフォルトはunrestrictedです。chmods これらのファイルおよびフォルダでが許可されないようにするには、このオプションをRestrictedに設定します。

NFSv4イジョウ

NFSv4では、アグレッシブなローカルクライアントキャッシュを提供する委譲の概念が導入されています。アグレッシブなローカルクライアントキャッシュはNFSv3のアドホックキャッシュとは異なります。委譲には、読み取りと書き込みの2種類があります。委譲は、パフォーマンスの向上よりもキャッシュの正確さを重視します。委譲を機能させるには、サポートされているUNIXクライアントと、NetAppコントローラでNFS 4.xバージョン固有の委譲オプションが有効になっている必要があります。これらのオプションはデフォルトで無効になっています。

サーバがファイル全体またはその一部をクライアントに委譲することを決定した場合、クライアントではそのファイルがローカルにキャッシュされ、サーバへの追加のRPC呼び出しが回避されます。これにより、ファイルの情報を取得するためのサーバへの要求が少なくなるため、読み取り委譲の場合のGETATTR呼び出しが少なくなります。ただし、委譲ではメタデータがキャッシュされないため、ファイル数の多いワークロードには、ストリーミングファイルワークロードほど大きなメリットはありません。

読み取り委譲は多数のクライアントに許可できますが、ファイルへの新たな書き込みによって委譲が無効になるため、書き込み委譲は一度に1つのクライアントにしか許可できません。有効な理由があれば、サーバは委譲をリコールすることができます。サーバは、2つのシナリオでファイルの委譲を決定します。1つはクライアントからの確認済みコールバックパスで、必要に応じてサーバが委譲のリコールに使用します。もう1つは、クライアントがファイルのOPEN関数を送信したときです。

読み取り委譲または書き込み委譲を使用する理由

委譲を使用すると、特定のアプリケーションの読み取りおよび書き込みパフォーマンスを向上させることができます。たとえば、同じクライアントまたは複数のクライアントにまたがる1つ以上のファイルからの読み取りが多数あり、GETATTRやLOOKUPなど大量のメタデータ処理を生成するWebアプリケーションは、パフォーマンスと応答時間を向上させるために、ストレージシステムからの読み取り委譲を要求できます。ファイル全体または特定の範囲のバイトをクライアントのローカル メモリに委譲することによって、メタデータ処理を目的としたネットワーク経由での追加のRPC呼び出しが回避されます。

委譲中にクライアントによってファイルオフセットまたはバイトオフセットが書き換えられた場合、委譲はリコールされます。委譲のリコールは更新内容を取得するためには必要な処理ですが、読み取りパフォーマンスに影響します。
したがって、単一のライターアプリケーションには書き込み委譲を使用する必要があります。読み取り委譲および書き込み委譲はI/Oパフォーマンスを向上させますが、どの程度向上するかは、クライアント ハードウェアやオペレーティング システムによって異なります。たとえば、メモリ容量の少ないクライアント プラットフォームでは、委譲がうまく処理されません。

委任を使用しているかどうかを確認するにはどうすればよいですか。

委譲を有効にし、クライアントでサポートされている場合は、それらが使用されます。委譲が使用されていることを確認するには、をオンにします `vserver locks show -type delegation`。

```
cluster::*> vserver locks show -type delegation

Vserver:  DEMO
Volume   Object Path                LIF         Protocol Lock Type    Client
-------- ---------------------------------- --------------- ------------ --------------- --------------
flexgroup_16
         /flexgroup_16/files/topdir_82/subdir_268/file3
                                    data        nfsv4.1   delegation -
                Delegation Type: write
         /flexgroup_16/files/topdir_27/subdir_249/file2
                                    data        nfsv4.1   delegation -
                Delegation Type: write
```

パフォーマンス統計を使用して委譲を確認することもできます。

```
cluster::*> statistics show -object nfsv4_1 -counter *del*

Object: nfsv4_1
Instance: DEMO
Start-time: 4/9/2020 15:05:44
End-time: 4/9/2020 15:31:13
Elapsed-time: 1529s
Scope: cluster
Number of Constituents: 2 (complete_aggregation)
    Counter                                                     Value
    ------------------------------------------ ------------------------------------------
    delegreturn_avg_latency                                  1366us
    delegreturn_percent                                         4%
    delegreturn_success                                      311465
    delegreturn_total                                        311465
```

NFSv4ロック

NFSv4クライアントの場合、ONTAPはNFSv4のファイルロックメカニズムをサポートしており、すべてのファイルのロック状態がリースベースモデルで維持されます。RFC 3530に従って、ONTAPはNFSクライアントが保持するすべての状態に対して1つのリース期間を定義します。この定義された期間内にクライアントがリースを更新しない場合は、クライアントのリースに関連付けられたすべての状態がサーバによって解放される可能性があります。クライアントはファイルの読み取りなどの操作を実行して、リースを明示的または暗黙的に更新できます。さらに、ONTAPは猶予期間を定義しています。猶予期間とは、サーバリカバリ中にクライアントがロック状態を再要求しようとする特別な処理期間のことです。

ロックは、ONTAPによってリースベースでクライアントに発行されます。デフォルトでは、サーバは各クライアントのリースを30秒ごとにチェックします。クライアントのリブートでは、クライアントが再起動後にサーバから有効なロックをすべて再要求できます。サーバがリブートした場合、サーバを再起動しても、デフォルトの猶予期間である45秒（ONTAPで最大90秒に調整可能）の間、クライアントに対して新しいロックは問題されません。この期間が過ぎると、要求するクライアントにロックが発行されます。30秒のリース期間はアプリケーションの要件に基づいて調整できます。NFSロックの管理については、製品ドキュメントの「ファイルロックの管理」を参照してください。

表8）NFSリース期間と猶予期間

期間	定義*
リース期間	ONTAPがクライアントに解除不能なロックを付与する期間
猶予期間	サーバリカバリ中にクライアントが自身のロック状態をONTAPに再要求する期間。

*詳細については、RFC 3530を参照してください。

NFSv4ロック リース期間の指定

NFSv4ロックリース期間（ONTAPがクライアントに解除不能なロックを付与する期間）を指定するには、 `-v4-lease-seconds` オプションを変更します。デフォルトでは、このオプションは30に設定されています。このオプションの最小値は10です。このオプションの最大値はロック猶予期間です。ロック猶予期間は `locking.lease_seconds` オプションで設定できます。

NFSv4.xリファーラル

最初のNFSマウント要求時に、NFSリファーラルがクライアントをSVM内の別のLIFに転送します。以降、NFSv4.xクライアントは、このリファーラルを使用して、アクセスを参照パス経由でターゲットLIFへ転送します。リファーラルは、データボリュームが存在するノード上のSVMにLIFがある場合に発行されます。つまり、クラスタ ノードが他のノード上のボリュームに対するNFSv4.x要求を受け取った場合、そのクラスタ ノードはLIF経由でそのボリュームのローカル パスを参照することになります。その結果、クライアントはダイレクト パスを使用してより速くデータにアクセスできるようになり、クラスタ ネットワーク上の余分なトラフィックも回避されます。

キノウ

マウント要求が送信される際には、要求は通常のNFSv4.xマウントとして処理します。ただし、DH LOOKUP呼び出しが行われると、サーバ（NetAppクラスタ）はGETFHステータスをにして `NFS4ERR_MOVED` クライアントに、要求されているLIFが存在する場所にアクセス中のボリュームが存在しないことを通知します。次に、サーバはクライアントにLOOKUP呼び出しを送信し、 `fs_location4 value`データボリュームが存在するノードのIPを（を使用して）通知します。これは、クライアントがDNS名とIPのどちらを使用してマウントしているかに関係なく機能します。ただし、クライアントは、サーバからクライアントに返されるIPではなく、指定されたIPにマウントされていることを報告します。

あるボリュームが別のノード上の別のアグリゲートに移動した場合、ボリュームがローカルである必要がある場合は、NFSv4.xクライアントでファイルシステムを手動でアンマウントおよび再マウントする必要があります。再マウントすると、クライアントがボリュームの新しい場所に参照されるようになります。手動でマウント / アンマウント処理を実行しなくてもクライアントが新しい場所のボリュームにアクセスできなくなることはありませんが、その場合I/O要求はリモート パスを経由します。
ただし、リモートI/O要求が環境に与える影響は、クライアントの再マウントに十分な大きさではない可能性があります。これはシステム停止を伴う処理である可能性があります。クライアントの再マウントは、状況に応じて決定する必要があります。

注： NFSv4.xリファーラルはRHEL 5.1（2.6.18-53）で導入されましたが、NetAppでは、NFSリファーラルで2.6.25より前のカーネルを使用しないこと、および1.0.12より前のバージョンのnfs-utilsは使用しないことを推奨しています。

ボリュームが他のボリュームの下でジャンクションされている場合、リファーラルはマウントされているボリュームをローカル ボリュームとして参照します。例：

- クライアントがvol2をマウントしようとしています
- vol2のジャンクションは/vol1/vol2です。
- vol1はnode1に、vol2はnode2に存在します。
- マウントがcluster：/vol1/vol2に対して行われます。
- リファーラルは、ホスト名クラスタのDNSから返されたIPアドレスに関係なく、node2に存在するLIFのIPアドレスを返します。
- マウントでは、node2のvol2に対してローカルなLIFを使用します。

クライアントが混在する環境でリファーラルをサポートしていないクライアントがある場合は、 `-v4.0-referrals` オプションを有効にしないでください。オプションが有効になっていて、リファーラルをサポートしていないクライアントがサーバからリファーラルを受け取った場合、そのクライアントはボリュームにアクセスできず、エラーが発生します。リファーラルの詳細については、RFC 3530を参照してください。

NFSv4.xのステートレス移行-Oracle dNFS

また、ONTAP 8.1以降ではNFSv4リファーラルによってNFSv4ステートレス移行がサポートされるようになり、Oracle dNFSのみがサポートされます。

移行は、クライアントのアクセスを中断することなくサーバ間でファイルシステムを移動できるNFSv4.xの機能です。移行を有効にするには、`-v4-fsid-change` NFSサーバでリファーラルとオプションを有効にする必要があります。移行はdiagレベルのオプションです。移行を有効にする場合、以下の要件を満たしていることが前提となります。

- SVMのNFSv4.xサーバにアクセスするすべてのクライアントがステートレスである。
- SVMのNFSv4.xサーバにアクセスするすべてのクライアントが移行をサポートしている。
- NFSv4.xクライアントが以下の機能を使用していない。
  - ロック
  - 共有予約
  - 委譲
  - ファイル アクセスのためのOPEN
- NFSv4.xクライアントが以下の機能を使用している。
  - READ、WRITE、およびSETATTRにすべてのビットが0の特殊なstateidが設定されている
  - OPENはファイルを作成するときだけで、作成後はすぐに閉じる
- NFSv4.xクライアントがNFSサーバで状態を確立していない。NFSの移行

のサポートは、ONTAPの次のシナリオで役立ちます。

- ボリューム移動
- LIFの移行とフェイルオーバー

表9）リファーラル、移行、pNFSの比較

	リファーラル	ステートレス移行	pNFS
リダイレクトのタイミング	マウント時	任意の操作（I/Oとメタデータ）	I/Oのみ（読み取り、書き込み）
リダイレクトされるトラフィック	すべてのトラフィック	すべてのトラフィック	I/Oのみ（読み取り、書き込み）
ユースケース	自動マウント	Oracle dNFS	I/Oでのデータの局所性を保証
欠点	マウント時のみ	ステートレス処理のみ（ロック状態なし）	非I/Oトラフィックがリダイレクトされない

NFSv4.xでのSnapshotコピー

NFSv3では `.snapshot`、デフォルトでディレクトリがクライアントに表示されます。（`.snapshot` NFSv3でのディレクトリの非表示については、「Snapshotコピーの非表示」を参照してください）。NFSv4.xではマウントプロトコルを使用しないため、`.snapshot` ディレクトリは表示されません。ただし、NFSv4.xマウント内のどこからでもアクセスできます。

NFSv4.xを使用してSnapshotコピーにアクセスする場合は、`.snapshot` ディレクトリに手動で移動するだけです。

```
# ls -la /nfs3 | grep snapshot
drwxrwxrwx 16 root                  root                  4096 Mar 31 14:05 .snapshot

# ls -la /nfs4 | grep snapshot
#
# ls -la /nfs4/.snapshot
total 64
drwxrwxrwx 16 root root   4096 Mar 31 14:05 .
drwxr-xr-x 14 root root   4096 Apr 11 2018 ..
drwx--x--x 8 root daemon 4096 Jan 19 2017 base
drwxrwxrwx 11 root root   4096 Jul 10 2017 clone_home_clone.0
drwxr-xr-x 14 root root   4096 Apr 11 2018 daily.2020-03-30_0010
drwxr-xr-x 14 root root   4096 Apr 11 2018 daily.2020-03-31_0010
drwxr-xr-x 14 root root   4096 Apr 11 2018 hourly.2020-03-31_0905
drwxr-xr-x 14 root root   4096 Apr 11 2018 hourly.2020-03-31_1005
```

負荷共有ミラーでのNFSv4.xの動作

ONTAPの負荷共有ミラーは、ボリュームの耐障害性を確保するために、SnapMirrorを使用して（負荷共有タイプを指定して）クラスタ内の複数のノードにレプリケートされるボリュームの読み取り専用コピーです。負荷共有ミラーはデータボリュームではサポートされていませんが、代わりにネームスペースのルートを含むSVMルートボリュームの保護に使用されます。これについては、「ネームスペースの保護」で詳しく説明します。

ただし、NFSv4.x処理ではボリュームのマウント時に負荷共有ミラーデスティネーションが利用されず、ソースボリュームのファイルハンドルが使用されます。そのため、負荷共有ミラーでは、SVMルートボリュームの障害に対する保護は提供されません。

NFSv4.1

NFSv4.1は、NFSv4のマイナーバージョン更新とみなされます。ここでは、NFSv4.1の仕様について説明します。前のセクションでは、NFSv4.0と、NFSv4.0とNFSv4.1の両方に該当するトピック（本ドキュメントではNFSv4.xで説明）について説明しました。

NFSv4.1を有効にしてNFSv4.0を無効にすることができます。この方法は、何らかの理由でクライアントがNFSv4.0を使用できないようにする場合に推奨します。

NFSv4.1を使用してクライアントをマウントするには、クライアントがNFSv4.1をサポートしている必要があります。NFSv4.1のサポート状況はクライアントのベンダーに確認してください。NFSv4.1のマウントには通常、`minorversion` マウントオプションを使用しますが、新しいLinuxカーネルでは、サポートされている最も高いNFSバージョンが自動ネゴシエーションされます。

例：

```
# mount -o nfsvers=4,minorversion=1 NFSSERVER:/unix /unix
```

NFSv4.1の機能

NFSv4.1では、RFC 5661で規定されているように、NFSv4プロトコル標準に多数の新機能が導入されています。これらの違いについては、RFCのセクション1.8で説明します。

一部の機能は必要に応じて記載されています。つまり、RFC標準に準拠するためには、この機能をNFSサーバに実装してサポートする必要があります。その他の機能は、推奨機能またはオプション機能としてリストされており、NFSサーバでアドホックにサポートされていますが、RFCへの準拠を主張するために必要な機能ではありません。たとえば、pNFSはNFSv4.1のオプション機能としてリストされており、ONTAPでサポートされていますが、NFSセッショントランキングとディレクトリ委譲（オプション機能も含む）は、現在ONTAPではサポートされていません。

パラレルネットワークファイルシステム

Parallel NFS（pNFS）は、NFSバージョン4.1標準の一部です。従来のバージョンであるNFS 3、4、4.1では、メタデータとデータは同じI/Oパスを共有します。pNFSは、メタデータとデータを異なるI/Oパスで処理します。クライアントからのすべてのメタデータアクティビティはメタデータサーバ（MDS）が処理し、データサーバがデータアクセスのためのダイレクトパスを提供します。

RFC 5661で説明されているように、

> パラレル データ アクセスはリコール可能なオブジェクト（「レイアウト」と呼ばれます）によって制御され、このレイアウトはプロトコル ロック モデルに統合されます。クライアントはデータ ストレージ プロトコル（NFSv4.1またはその他のプロトコル）を使用して、レイアウトによって指定されたデータ サーバのデータセットに、データ アクセス要求を転送します。

pNFSの場合、NetAppはpNFSをサポートし、RFC仕様に準拠しているすべてのクライアントをサポートします。pNFSオプションはデフォルトで有効になっていますが、NFSv4.1のサポートも有効になっている場合にのみアクティブになります。

pNFSの仕組み

pNFSは、サーバ（ONTAPで実行されているNFSサーバ）によって生成され、クライアントに送信されるデバイスの概念を定義します。これにより、クライアントはデータを見つけやすくなり、そのデータに対してローカルなパスで要求を直接送信することができます。ONTAPは、フレキシブルボリュームごとに1つのpNFSデバイスを生成します。メタデータパスは変更されないため、メタデータ要求がリモートである可能性があります。ONTAP pNFS環境では、すべてのデータLIFがNFSサーバとみなされるため、pNFSは、各ノードがNFS SVMごとに少なくとも1つのデータLIFを所有している場合にのみ機能します。そうしないと、pNFSの利点が無効になります。pNFSは、クライアントがどのIPアドレスに接続するかに関係なく、データの局所性を意味します。データLIFが停止した場合のpNFSの動作については、「pNFSとLIFの停止」を参照してください。

pNFSデバイスには、次の情報が含まれています。

- ボリューム コンスティチュエント
- コンスティチュエントのネットワーク上の場所

デバイス情報は、パフォーマンスを向上させるためにローカルノードにキャッシュされます。

クラスタ内のpNFSデバイスを表示するには、advanced権限で次のコマンドを実行します。

```
cluster::> set diag
cluster::*> vserver nfs pnfs devices cache show
```

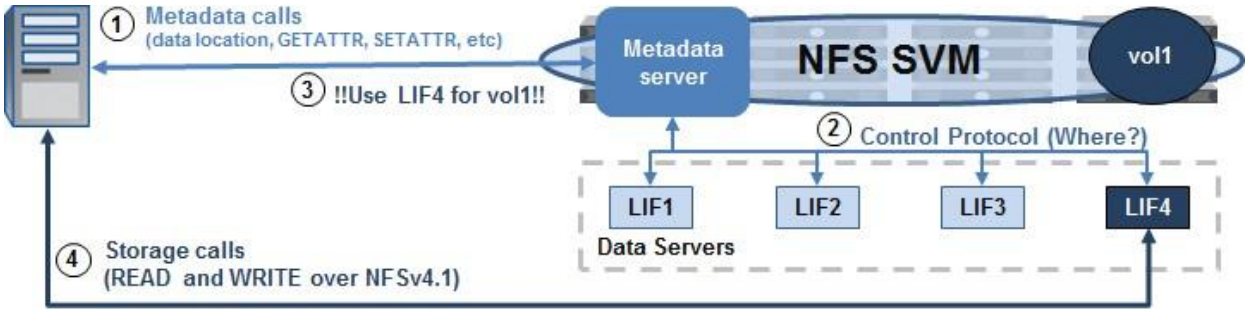
pNFSコンポーネント

pNFSには3つの主要なコンポーネントがあります。

- MDS：
  - データ トラフィック以外のすべてのトラフィック（GETATTRやSETATTRなど）を処理します。
  - クライアントにファイルの場所を通知するメタデータを保存します。
  - NetApp NFSサーバ上にあります。
- データサーバ：
  - ファイルデータを格納し、読み取り要求と書き込み要求に応答する
  - NetApp NFSサーバ上にあります。
  - inode情報もここにあります。
- クライアント

これらのコンポーネントは、3つの異なるプロトコルを利用します。制御プロトコルは、メタデータサーバとデータサーバの同期を維持する方法です。pNFSプロトコルは、クライアントとMDSの間で使用されます。pNFSは、ファイルベース、ブロックベース、およびオブジェクトベースのストレージプロトコルをサポートしますが、NetAppは現在ファイルベースのpNFSのみをサポートしています。

図8）pNFSのデータワークフロー



1. The client makes a data request to the cluster.
2. The metadata server works to find the location of the data if the location is not already cached.
3. The location of the data is returned to the client via the control path.
4. The client begins operations over the specified data LIF returned from the metadata server.

pNFSが使用されていることを確認するにはどうすればよいですか。

pNFSが使用されているかどうかを確認するには、統計カウンタを実行して `pnfs_layout_conversions` カウンタを確認します。の数 `pnfs_layout_conversions` が増えている場合は、pNFSが使用されています。

```
cluster::*> statistics show -object nfsv4_1_diag -counter pnfs_layout_conversions

Object: nfsv4_1_diag
Instance: nfs4_1_diag
Start-time: 4/9/2020 16:29:50
End-time: 4/9/2020 16:31:03
Elapsed-time: 73s
Scope: node1

    Counter                                                     Value
    -------------------------------------------- ----------------------
    pnfs_layout_conversions                                      4053
```

NFSv3とNFSv4.xを使用したpNFSの基本的な比較テストについては、本ドキュメントの次のセクションを参照してください。

- 『パフォーマンスの比較：nconnectとpNFSを使用したNFSv3とNFSv4の比較』
- 「NFSv3と NFSv4.x -パフォーマンス比較」
- 「異なるTCP最大転送ウィンドウサイズのパフォーマンスの例」

注: pNFSを使用する場合は、次のようなバグを回避するために、利用可能な最新のクライアントとONTAPリリースを使用してください。SU323: Linuxディストリビューションでの pNFS I/O中にデータが破損する可能性があります。

クライアントをブラックリストに登録する方法

NFSサーバでpNFSを有効にすると、NFSv4.1以降を使用してマウントするすべてのクライアントでpNFSが利用されます。場合によっては、pNFSではなく基本的なNFSv4.xのみを使用するように設定することもできます。

次のコマンドを使用すると、NFSv4.1のマウントが確立されたときに、クライアントでpNFSが使用されているかどうかを確認できます。`nfs_layout_nfsv41_files` pNFSが使用されているかどうかが表示されます。

```
# lsmod | grep nfs
nfs_layout_nfsv41_files    32768 1
```

```
nfsv4                  790528  3 nfs_layout_nfsv41_files
dns_resolver            16384  2 cifs,nfsv4
nfsv3                   49152  1
nfs_acl                 16384  1 nfsv3
nfs                    360448  5 nfsv4,nfs_layout_nfsv41_files,nfsv3
lockd                  122880  2 nfsv3,nfs
fscache                385024  2 nfsv4,nfs
sunrpc                 479232  28
```

pNFSを使用する特定のクライアントをブラックリストに登録するには、pNFSモジュールが `nfs_layout_nfsv41_files` クライアントにロードされないように `/etc/modprobe.d/nfs_layout_nfsv41_files-blacklist.conf` ファイルをブラックリストに登録します。

例：

```
cat /etc/modprobe.d/nfs_layout_nfsv41_files-blacklist.conf
blacklist nfs_layout_nfsv41_files
```

注： 変更を有効にするにはリブートが必要です。

モジュールを表示して、NFSv4.1のマウントが確立されたときにモジュールが無効またはロードされていないことを確認するには、次の手順を実行します。

```
# lsmod | grep nfs
nfsv4                  584056  3
dns_resolver            13140  1 nfsv4
nfs                    262045  16 nfsv4
fscache                 64980  2 nfs,nfsv4
nfsd                   351321  13
auth_rpcgss             59415  2 nfsd,rpcsec_gss_krb5
nfs_acl                 12837  1 nfsd
lockd                   98048  2 nfs,nfsd
grace                   13515  2 nfsd,lockd
sunrpc                 358543  46 nfs,nfsd,rpcsec_gss_krb5,auth_rpcgss,lockd,nfsv4,nfs_acl
```

モジュールのブラックリスト化の詳細については、How do I prevent a kernel module from automatically loading? を参照してください。

NFSv4.1の委譲

NFSv4.1の委譲はNFSv4.0の委譲と非常によく似ていますが、v4.0ではなくv4.1プロトコルの一部です。表10に、NFSv4.1で新たに追加された機能と、NFSv4.0と比較した場合の環境のメリットを示します。これらの追加機能の詳細については、RFC 5661のセクション10.2を参照してください。

表10）NFSv4.1の委譲のメリット

NFSv4.1の委譲機能	NFSv4.0の委譲と比較した場合のメリット
EXCHANGE_IDを使用	NFSv4.0ではSETCLIENTIDが使用されていました。EXCHANGE_IDはSETCLIENTIDに置き換わるもので、他のクライアント処理が行われる前にクライアントIDを割り当てることができるようにします。RFC 5661に記載されているように、クライアントIDが確立される前に実行できるNFSv4.1の処理は、クライアントIDを確立するために必要な処理だけです。
コールバックがフォアチャネルと同じTCP接続を使用	NFSv4.0では、コールバックがフォアチャネルと異なるTCP接続を使用します。コールバックに同じTCP接続を使用することで、委譲のパフォーマンスが向上し、ファイアウォールとの親和性も高くなります。
新しいOPEN要求オプション： • OPEN4_SHARE_ACCESS_WANT_DELEG_MASK	NFSv4.1では、クライアントによる委譲の獲得をNFSv4.0よりも細かく制御できます。

NFSv4.1の委譲機能	NFSv4.0の委譲と比較した場合のメリット
• OPEN4_SHARE_ACCESS_WANT_NO_PREFERENCE • OPEN4_SHARE_ACCESS_WANT_READ_DELEG • OPEN4_SHARE_ACCESS_WANT_WRITE_DELEG • OPEN4_SHARE_ACCESS_WANT_ANY_DELEG • OPEN4_SHARE_ACCESS_WANT_NO_DELEG	これらの新しいオプションによって、より多くのOPENシナリオに対応できるようになり、委譲の発行時または再要求時の問題を防止できます。

NFSv4.1セッション

RFC 5661によると、次のようになります。

> セッションは、動的に作成される長時間有効なサーバ オブジェクトです。クライアントによって作成され、1つ以上のトランスポート接続で一定期間にわたって使用されます。その役割は、あるクライアントインスタンスに属する接続に対するサーバの状態を維持することです。この状態は、接続自体とはまったく関係がなく、接続の有無にかかわらず存在します。クライアントには1つ以上のセッションを関連付けることができます。それらのセッションに接続が関連付けられると、そのクライアントのクライアントIDに関連付けられたセッションのいずれかを使用してクライアントに関連する状態にアクセスできるようになります。あるクライアントIDのどのセッションにも長時間接続が関連付けられなかった場合は、ロック、オープン、委譲、レイアウトなどのオブジェクトの有効期限が切れます。セッション サーバは、クライアントがサーバ上の関連するクライアント状態にアクセスする手段を提供するオブジェクトであり、その状態にアクセスする物理的手段とは無関係です。
>
> 1つのクライアントが複数のセッションを作成できます。1つのセッションを複数のクライアントで使用することはできません。

SNIAから：

> セッションNFSv4.1には、セッションとpNFSという2つの主要な機能があります。セッションは、NFSセマンティクスに正確さと簡易性のメリットをもたらします。NFSv4の正確性を高めるために、NFSv4.1セッションには「exact-once」というセマンティクスが導入されています。これは、冪等でない操作（つまり、ファイル名変更操作など、2回以上実行すると異なる結果が返される操作）をサポートする場合に重要です。NFSの場合と同様に、ファイルシステムとストレージが信頼性の低い通信リンクによって分離されている場合、このような操作が重要な実用的な問題となります。サーバは、クライアントと一致した1つ以上のセッション状態を保持します。セッションは、クライアントに属する接続に対するサーバの状態を保持します。クライアントは、サーバへの要求が実行されたこと、および2回以上実行されることがないことを保証できます。セッションは、サーバから開始される非同期コールバックを導入したNFSv4委譲の概念を拡張したものです。クライアントはサーバへの接続のセッション要求を開始できます。WANベースのシステムでは、ファイアウォールを介した運用が簡素化されます。

NFSv4.1セッショントランキング

ONTAPのNFSv4.1サーバは現在、セッショントランキング（マルチパス）機能をサポートしていません。この機能は、VMware環境でよく使用されます。

NFSv4.2

NFSv4.2は利用可能な最新のNFSv4.xバージョンであり、RFC-7862でカバーされています。ONTAP 9.8では、NFSv4.2プロトコルの基本的なサポートが導入されました。ONTAP 9.9ではラベル付きNFSのサポートが追加されましたが、その他の補助機能は現在サポートされていません。NFSv4.2には独自の有効/無効オプションはありませんが、`-v4.1` ONTAPのNFSサーバオプションを使用してNFSv4.1を有効にすると有効/無効になります。NFSv4.2をサポートするクライアントでは、指定しない場合、mountコマンドでサポートされる最上位バージョンのNFSがネゴシエートされます。それ以外の場合は、 `minorversion=2` マウントオプションを使用します。NFSv4.1とNFSv4.2では、パフォーマンスに違いはありません。

ラベルがNFS 4.2

ONTAP 9.9.1では、ラベル付きNFSと呼ばれるNFSv4.2の機能がサポートされています。この機能を使用すると、SELinuxラベルと強制アクセス制御（MAC）を使用してファイルやフォルダへのきめ細かなアクセスを管理できます。これらのMACラベルはファイルおよびフォルダに格納され、UNIX権限およびNFSv4.x ACLと連携して機能します。この機能を有効または無効にするには、次のadvanced権限オプションを使用します。

```
[-v4.2-seclabel {enabled|disabled}] - NFSV4.2 Security Label Support (privilege: advanced)
This optional parameter specifies whether to enable security labels for NFSv4.2. The default
setting is disabled at the time of creation.
```

ラベル付きNFSがサポートされるため、ONTAPはNFSクライアントのSELinuxラベル設定を認識して認識できるようになりました。

ラベル付きNFSは RFC-7204でカバーされています。

主なユースケース

- 仮想マシン（VM）イメージのMACラベル付け
- 公共機関のデータセキュリティ分類（シークレット、トップシークレットなど）
- セキュリティコンプライアンス
- ディスクレスLinux

このリリースでは、ONTAPで次の強制モードがサポートされています。

- 制限されたサーバーモード。ONTAPはラベルを強制することはできませんが、ラベルを保存および送信することはできます。

  注: MACラベルを変更する機能は、強制するクライアントによっても異なります。

- ゲストモード。クライアントにNFS対応（v4.1以前）のラベルが付けられていない場合、MACラベルは送信されません。ONTAPは現在、フルモード（MACラベルの保存と適用）をサポートしていません。

ネーム サービス

エンタープライズNAS環境では、毎日、何千ものクライアント、ユーザ、グループがストレージシステムとやり取りしています。これらのクライアント、ユーザ、グループには、すべてのNASクライアントで一貫した管理が必要です。クライアントAのuser1は、クライアントBのuser1と同じにしないでください。また、クライアントAとクライアントBは、同じホスト名またはIPアドレスを使用しないでください。

そこで、ネームサービスが登場します。

DNS

DNSサーバを使用すると、IPアドレス、ホスト名、ビジネスクリティカルなサービスレコードを一元的に作成および管理できます。すべてのクライアントとストレージシステムが同じDNS設定を参照している場合は、ホスト名<-> IPマッピングの一貫性が確保され、数千ものローカルファイルを管理する必要はありません。

DNSは、次のような多くのアプリケーションやネットワークサービスで重要です。

- Kerberos
- LDAP
- Active Directory

特にKerberosとLDAPを使用する場合は、NFS環境でDNSを使用することを強く推奨します。

動的DNS

DNSでは、IPアドレスが追加、削除、または変更されたときに、クライアントからDNSサーバにDNS更新を送信することができます。この機能は、DNSに必要な管理オーバーヘッドの量を削減しますが、DNSサーバでサポートされている場合にのみ可能です。

ONTAPは、データLIFが動的DNSを介してDNSサーバに更新を送信するための手段を提供します。これは `vserver services name-service dns dynamic-update` コマンドで管理します。

DNSロード バランシング

場合によっては、ホスト名が単一のIPアドレスだけでなく、複数のIPアドレスのフロントエンドになることもあります。ONTAPでは、1つのSVMに複数のデータLIFが設定される場合があります。NFSクライアントは通常、これらのネットワークインターフェイスにDNSホスト名を使用してアクセスし、複数のIPアドレス間で接続の負荷を分散します。ONTAPでは、DNSとの接続の負荷分散方法をいくつかサポートしています。

- オフボックスDNS（レコードを介したラウンドロビン）
- 内蔵DNS（ONTAP DNSサーバへのDNS転送/委譲）
- サードパーティ製ロードバランサ（ハードウェアまたはソフトウェアゲートウェイ）

オンボックスDNSかオフボックスDNSか？

ONTAPは、内蔵DNSサーバを使用してDNSクエリを処理する方法を提供します。この方法では、ノードのCPUとスループットを考慮して、NASアクセス要求を処理するのに最適なデータLIFを特定します。

- 外部DNSを設定するには、DNS管理者が、データLIFへのラウンドロビンアクセスを提供する外部DNSサーバ上に同じ名前のAネームレコードを複数作成します。
- マウントストームのシナリオを作成するワークロードの場合、ONTAP内蔵DNSサーバが適切に維持およびバランス調整できないため、外部DNSを使用することを推奨します。

詳細については、TR-4523：『DNS Load Balancing in ONTAP』を参照してください。

アイデンティティ管理ネームサービス

アイデンティティ管理のために、LDAPおよびNISは、ネットグループ機能に加えて、ユーザとグループ用の中央リポジトリを提供します。これらの一元化されたサービスにより、クライアントとサーバは同じ情報を維持し、NASファイルシステムにアクセスする際に予測可能で一貫性のあるIDを確保できます。

ONTAPはネームサービスでLDAPとNISの両方をサポートしていますが、セキュリティとレプリケーションのサポートのためにNISよりもLDAPを推奨します。

LDAP

ユーザ、グループ、およびネットグループのID管理には、LDAPサーバを使用することを推奨します。LDAPは、ネームサービスのソースをNFSクライアントとサーバ全体で一元化するだけでなく、SSLまたはKerberosを使用したセキュアなバインドと検索を介してLDAPパケットを暗号化し、通信を保護する方法も提供します。NISサーバは、デフォルトではこの機能をサポートしていません。

さらに、LDAPサーバは、特にActive Directoryを使用してUNIX ID管理を行う場合に、複数のサーバ間で情報を複製するための簡単な方法を提供します。ONTAP NAS環境で使用するLDAPの設定の詳細については、TR-4835：『How to Configure LDAP in ONTAP』を参照してください。

NIS

ONTAPでは、ユーザ、グループ、およびネットグループのネームサービスにNISデータベースを使用することもできます。ONTAPは'ypserv呼び出しを使用した検索に標準のnis*.byname機能を使用しますNISの標準機能を利用するすべてのNISサーバ（Windows Active Directoryを含む）を検索に使用できます。

ONTAPでは、チャット可能なNIS環境に対して、ローカルNIS group.bynameおよびnetgroup.byname機能（NISスレーブと同様）を有効にすることもできます。これにより、有効にした場合のネットワークおよびNISサーバの全体的な負荷を軽減できます。

ローカルファイル

ローカルファイル（passwd、group、netgroupなど）もネームサービスソースとしてサポートされます。ONTAPでは、ストレージ管理者は、ONTAPコマンドを使用してファイルを作成するか（UNIXユーザおよびグループの作成）、 `load-from-uri` コマンドを使用してサーバからフラットファイルをインポートできます。ONTAP SVMでは、ローカルUNIXユーザおよびグループに対してデフォルトで最大64、000個のエントリがサポートされます。

ローカルファイルがプライマリネームサービスになり、64,000を超えるエントリが必要になる場合は、拡張/ファイルのみモードを有効にすることをお勧めします。

拡張モード/ファイル専用モード

ONTAP 9.1以降では、ローカルユーザとローカルグループの拡張モード/ファイルオンリーモードを使用できます。ストレージ管理者は、diagレベルのネームサービスオプションを有効にし、 `load-from-uri` 機能を使用してクラスタにファイルをロードしてより大きな数を指定することで、ローカルユーザとローカルグループの制限を拡張できます。 ユーザとグループの数拡張モード/ファイル専用モードでは、ネームサービスサーバやネットワークなどに外部の依存関係が不要になるため、ネームサービス検索のパフォーマンスが向上します。ただし、ファイル管理によってストレージ管理のオーバーヘッドが増大し、人為的ミスの可能性が高まるため、このパフォーマンスにはネームサービスの管理が容易になりません。また、ローカルファイル管理はクラスタごとに行う必要があるため、複雑さがさらに増します。

ユーザとグループに対してこのオプションを有効にするには、 `vserver services name-service unix-user file-only` コマンドと `vserver services name-service unix-group file-only` コマンドを実行します。

モードを有効にしたら、次のコマンドを実行してURIからユーザとグループのファイルをロードします。

```
cluster::*> vserver services name-service unix-user load-from-uri
```

メモ： ユーザの場合は10MB、グループの場合は25MBを超えるファイルをロードする場合は、`-skip-file-size- check` オプションを使用します。

ファイルのみモードを使用している場合、ユーザおよびグループに対する個 々 の操作は許可されません。この構成は、現在、NetApp MetroClusterまたはSVMディザスタリカバリ（SVM DR）のシナリオではサポートされていません。

ファイル専用モードを使用している場合でも、外部ネームサービスを使用できますか。

ファイルのみモードでは、LDAPまたはNISをネームサービスとして使用できないわけではありません。つまり、ローカルユーザとローカルグループは（レプリケートされたデータベースエントリではなく）ファイルのみで管理されます。ファイルのみモードが有効になっている場合でも、LDAPおよびNIS検索は正常に機能します。

デフォルトのローカルユーザ

SVMのセットアップまたはSystem Managerを使用してSVMを作成すると、デフォルトのローカルUNIXユーザおよびグループが、デフォルトのUIDおよびGIDとともに作成されます。

次の例は、これらのユーザとグループを示しています。

```
cluster::> vserver services unix-user show -vserver vs0
              User            User   Group Full
```

```
Vserver         Name              ID      ID      Name
--------------- ----------------- ------- ------- ----------------------------------------------------------------------
nfs             nobody            65535 65535 -
nfs             pcuser            65534 65534 -
nfs             root              0       0       -

cluster::> vserver services unix-group show -vserver vs0
Vserver         Name                  ID
--------------- --------------------- ---------
nfs             daemon                1
nfs             nobody                65535
nfs             pcuser                65534
nfs             root                  0
```

注: ファイル専用モードを使用する場合は、クラスタの管理に使用するファイルに上記のユーザが存在していることを確認してください。ファイル専用モードを有効にすると、アップロードされたファイルにデフォルトユーザが含まれていない場合、デフォルトユーザは削除されます。

ローカルユーザへの影響

ファイル専用モードが有効になっている場合、root、pcuserno、body ロードされているファイルにユーザが含まれていない場合、およびのデフォルトのローカルユーザが削除されます。ファイル専用モードを使用する場合は、passwd/groupファイルにローカルユーザとローカルグループを含めるようにしてください。

条件

次のセクションでは、ONTAPでローカルユーザとローカルグループを使用する場合の制限について説明します。これらの制限はクラスタ全体に適用されます。

表11）ONTAPでのローカルユーザとローカルグループの制限

	ローカルUNIXユーザ/グループ	拡張モードのユーザ/グループ
ローカルユーザおよびローカルグループの最大エントリ数	65,536	ユーザ数：400,000 グループ：15,000 グループメンバーシップ：3、000 SVM：6
拡張モードのユーザおよびグループの最大ファイルサイズ	該当なし	パスワードファイルのサイズ（ユーザ）：10MB * グループファイルのサイズ：25MB * *グループおよびpasswdファイルのサイズは上書きでき - skip-file-size-check ますが、ファイルサイズが大きい場合はテストされていません。

前述したように、ローカルUNIXユーザおよびグループの制限はクラスタ全体に適用され、これにはSVMが複数あるクラスタも該当します。したがって、クラスタにSVMが4つある場合は、各SVMの最大ユーザ数の合計が、クラスタの最大数に達している必要があります。

例：

- SVM1のローカルUNIXユーザ数は2,000
- SVM2のローカルUNIXユーザ数は40,000
- SVM3のローカルUNIXユーザ数は20
- この場合、SVM4で作成できるローカルUNIXユーザ数は23,516となります。

上限を超える数のUNIXユーザまたはグループを作成しようとすると、エラー メッセージが表示されます。

例：

```
cluster::> unix-group create -vserver NAS -name test -id 12345
```

```
Error: command failed: Failed to add "test" because the system limit of {limit number}
       "local unix groups and members" has been reached.
```

マルチプロトコルのNAS

ONTAPは、SVM内の同じデータセットへのマルチプロトコルNASアクセスをサポートしています。ONTAPのマルチプロトコルNASアクセスでは、ユーザとグループは、CIFS / SMBおよびNFSを使用してボリュームまたはqtreeにアクセスし、ユーザとグループのACLを活用し、ファイルとフォルダの所有権を必要に応じて設定できます。マルチプロトコルNASの詳細については、TR-4887：『Multiprotocol NAS in ONTAP - Overview and Best Practices』を参照してください。

qtree

ストレージ管理者は、qtreeを使用してONTAP UIまたはCLIからフォルダを作成し、ボリューム内のデータを論理的に分離できます。qtreeでは、独自のエクスポートポリシー、独自のセキュリティ形式、クォータ、および詳細統計を有効にすることで、データ管理を柔軟に行うことができます。

qtreeには複数のユースケースがあり、ホームディレクトリのワークロードに役立ちます。qtreeには、データにアクセスするユーザのユーザ名を反映した名前を付けることができ、動的共有を作成してユーザ名に基づいたアクセスを提供できるためです。

FlexGroupボリューム内のqtreeに関する詳細情報を次に示します。

- qtreeは、クライアントにはディレクトリとして表示されます。
- qtreeはボリュームレベルで作成できます。現在のところ、ディレクトリの下にqtreeを作成してサブディレクトリであるqtreeを作成することはできません。
- qtreeの作成と管理は、FlexVol qtreeの管理と同じ方法で行います。
- SnapMirrorを使用してqtreeをレプリケートすることはできません。現在、SnapMirrorはボリュームレベルでのみ実行されます。ボリュームを使用したレプリケーションをさらに細かく行う場合は、ジャンクションパスを使用します。
- ボリュームあたり最大4、995個のqtreeがサポートされます。クォータの監視と適用（FlexGroupボリュームのONTAP 9.5以降では適用）は、qtreeレベルまたはユーザレベルで適用できます。

qtreeとファイル移動

qtreeは、ONTAPでは一意のファイルシステムとみなされます。NASクライアントからはディレクトリのように見えますが、一部の処理は実際のディレクトリとは動作が異なることがあります。たとえば、同じボリューム内のqtree間でファイルを移動する場合などです。

複数のディレクトリにまたがるボリューム内でファイル移動を実行すると、ファイル名が新しい名前に変更されます。同じファイルシステム内でファイル移動が行われるため、ファイル移動は数秒で実行されます。

2つのqtree間でファイルの移動が発生すると、名前が変更されるのではなく、新しい場所にファイルがコピーされます。これにより、処理にはるかに時間がかかります。

これは、qtreeがFlexVolボリュームとFlexGroupボリュームのどちらに存在するかに関係なく発生する動作です。

qtreeのIDと名前変更の動作

継承されていないエクスポートポリシーをqtreeに適用すると、qtree間の操作を処理する際にNFSファイルハンドルがわずかに変更されます。ONTAPはNFS処理でqtree IDを検証します。この処理は、ソースフォルダまたはqtreeと同じボリューム内のqtreeとの間で移動する際のファイル名の変更や移動などに影響します。これはセキュリティ機能とみなされ、ホームディレクトリのシナリオなど、qtree間の不要なアクセスを防止できます。ただし、エクスポートポリシールールと権限を適用するだけでも同様の目的を達成できます。

たとえば、同じボリューム内のqtreeを移動したりqtree名を変更したりすると、「Access Denied」エラーが発生します。別のボリューム内のqtreeとの間で同じ移動または名前変更を行った場合、ファイルがコピーされます。ファイルのサイズが大きい場合は、移動処理に異常に長い時間がかかっているように見えることがありますが、ほとんどの移動処理は、同じファイルシステムまたはボリューム内での単純なファイル名変更であるため、ほぼ瞬時に完了します。

この動作はadvanced権限オプションで制御されます。詳細については、NetAppナレッジベースの記事「Permission denied while moving files between trees when nfs option '-validate-qtree-export'」を参照してください。

ナレッジベースの記事では、さまざまな操作の動作について説明しています。

```
Assuming that file permissions allow and that client is allowed by export policies to access both
source and destination volume/qtree, these are the current permutations with the 'validate-qtree-
export' flag enabled or disabled:

有効：
- Rename in same volume and qtree: SUCCESS
- Rename in same volume, different qtrees: EACCESS
- Rename between volumes where qtree IDs differ: EACCESS
- Rename between volumes where qtree IDs match: XDEV

無効：
- Rename in same volume and qtree: SUCCESS
- Rename in same volume, different qtrees: SUCCESS
- Rename between volumes where qtree IDs differ: XDEV
- Rename between volumes where qtree IDs match: XDEV
```

注： NFS3ERR_XDEVとNFS3ERR_ACCESSは RFC-1813で定義されています。

名前の変更やqtree間の移動の動作を変更するには、`-validate-qtree-export` をに変更します `disabled`。詳細については、「qtreeファイル操作のqtree IDの検証」を参照してください。

注： `-validate-qtree-export` オプションを無効にした場合、qtree間で名前変更が許可される以外には、既知の悪影響はありません。

qtreeエクスポートに対するファイルハンドルの影響

通常、クライアントに渡されるNFSエクスポートファイルハンドルのサイズは32バイト以下です。ただし、qtreeエクスポートでは、40バイトのファイルハンドルを作成するために数バイトが追加されます。ほとんどのクライアントでは、これは問題ではありませんが、古いクライアント（1996年に導入されたHPUX 10.20など）では、これらのエクスポートのマウントで問題が発生する可能性があります。qtreeエクスポートを有効にする前に、古いクライアント接続を別のテストSVMでテストするようにしてください。qtreeエクスポートを有効にしたあとにファイルハンドルの動作を変更する方法は現時点ではないためです。

同じNFSクライアント上の同じボリューム内の複数のqtreeのマウント

qtreeは実質的に独立したファイルシステムとして機能しますが、qtreeが同じボリュームに配置されている場合、クライアントとサーバ間のNFS通信では親ボリュームと同じMSID /ファイルハンドルが使用されます。その結果、NFSクライアントでqtreeが同じファイルシステムとして2回マウントされていることが確認され、各qtreeで実際に使用されているスペースに関係なく使用済みスペースが同じになることがあります。

たとえば、これら2つのqtreeは、異なるマウントポイントで同じクライアントにマウントされます。

```
# mount | grep qtree
10.193.67.214:/testvol/qtree1 on /mnt/qtree1 type nfs
10.193.67.214:/testvol/qtree2 on /mnt/qtree2 type nfs
```

どちらの場合も、ファイルをコピーする前に同じスペース使用量が表示されます。

```
# df -h | grep qtree
10.193.67.214:/testvol/qtree1  973G 2.0M 973G     1% /mnt/qtree1
10.193.67.214:/testvol/qtree2  973G 2.0M 973G     1% /mnt/qtree2
```

次に、3.8GBのファイルをqtree1にコピーします。両方のqtreeで同じスペースが使用されています。

```
# cp debian-8.2.0-amd64-DVD-1.iso /mnt/qtree1/
# df -h | grep qtree
10.193.67.214:/testvol/qtree1  973G 3.8G 970G     1% /mnt/qtree1
10.193.67.214:/testvol/qtree2  973G 3.8G 970G     1% /mnt/qtree2
```

この問題を回避するには、いずれかのqtreeにクォータを監視します。これを行うだけで、適切なスペース使用量が表示されます。

```
cluster::*> quota report -vserver NFS
Vserver: NFS

                                       ----Disk---- ----Files-----   Quota
Volume   Tree       Type    ID         Used Limit     Used   Limit   Specifier
-------- ---------- ------- ---------- -------- -------- -------- ------------
testvol  qtree1     tree    1
                                      3.73GB       -        2        -   qtree1
testvol  qtree2     tree    2            0B       -        1        -   qtree2
testvol             tree    *            0B       -        0        -   *

# df -h | grep qtree
10.193.67.214:/testvol/qtree1  973G 3.8G 970G     1% /mnt/qtree1
10.193.67.214:/testvol/qtree2 970G      0 970G    0% /mnt/qtree2
```

サブディレクトリのエクスポート

qtreeはNFS経由でエクスポートできます。NFSは、単一レベルのサブディレクトリパスを提供し、クライアントに固有のエクスポートポリシーとルールを定義します。ただし、個々のディレクトリにエクスポートポリシーとルールを適用することはできず、現在ONTAPではqtreeをボリュームレベルでしか作成できません。ディレクトリツリーの下位レベルのエクスポートが必要な環境では、ボリューム、qtree、およびジャンクションパスを組み合わせてサブディレクトリのエクスポートをシミュレートできます。ただし、ジャンクションパスの各レベルでは、クライアントがトラバーサルを許可するためにエクスポートポリシールールへの読み取りアクセスを許可する必要があるため、パス全体が保護されるわけではありません。

たとえば、次のようなサブディレクトリエクスポートを作成できます。

```
/volume1/qtree1/volume2/qtree2/volume3/qtree3
```

このパス内の各オブジェクトは、一意のポリシーとルールを使用してNFSクライアントにエクスポートできます。これらのフォルダのセキュリティレベルを高める必要がある場合は、NFSにNTFSセキュリティ形式/ACLまたはKerberosを使用することを検討してください。

ユーザおよびグループの所有者

ONTAP 9.8以降では `qtree create` 、ONTAP CLIでまたはを使用してqtreeのユーザおよびグループの所有者を設定できます `qtree modify`。以前のリリースでは、クライアントからNASプロトコルを使用して実行されていました。現時点では、CLIまたはREST APIでのみ使用できます。ZAPIまたはONTAP System Managerはサポートされません。

```
[ -user <user name> ]              User ID
[ -group <group name> ]            Group ID
```

NFSでのノンストップオペレーション

このセクションでは、ONTAPでのNFSでのノンストップオペレーション（NDO）、およびNFSクライアントでのNDO動作のシナリオについて説明します。場合によっては、NFSv3でも特定の計画的または計画外のイベントによってシステムが停止することがあります。NFSv3がステートレスプロトコルであるにもかかわらず、ロックやNFSサーバ側キャッシュなどの基盤となるメカニズムが、システム停止イベントの際に有効になる可能性があるためです。

再生/応答キャッシュ

NFS要求が冪等でない要求を2回試行しないようにするには、ONTAPの再生（または応答）キャッシュが重要です。非冪等リクエストは、データ構造を変更できるリクエストです。たとえば、ドキュメントを同時に2回読むことは無害であるため、偶発的な操作です。そのドキュメントを同時に2回編集することは非冪等な操作であり、ドキュメントを保護するためのロックが設定されていないと有害になる可能性があります。ONTAPの再生/応答キャッシュは、ネットワーク問題によってクライアントが同じ処理を再送信した場合に、どの処理がストレージに到達したかを追跡するのに役立ちます。キャッシュは、ストレージレイヤで再試行するのではなく、処理への応答に使用されます。

このキャッシュは、データ レイヤにボリュームとともに保管されます。このキャッシュが失われると、作成操作はE_EXISTで失敗し、削除操作はE_NOENTで失敗する可能性があります。表12 に、再生キャッシュが保持または失われるさまざまなシナリオを示します。これにより、処理の中断が決まります。

表12）再生/応答キャッシュのNDO動作

処理	結果（NFSv3およびNFSv4.x）
ボリューム移動	再生キャッシュはボリュームとともに移動される。
アグリゲートの再配置またはストレージのギブバック処理	再生キャッシュは失われる。
LIFの移行（同じノード）	再生キャッシュはそのまま維持される。
LIFの移行（別のノード）	再生キャッシュは失われる。
計画外のテイクオーバー	再生キャッシュは失われる。
計画的なテイクオーバー	再生キャッシュは失われる。

フアイルロツク

ファイル ロック メカニズムは、同じファイルに複数のユーザまたはアプリケーションが同時に書き込み目的でアクセスしないようにするために開発された機能です。NFSでは、NFSv3のNLMプロセスを使用するか、NFSv4.xプロトコルに組み込みのリースとロックの機能を使用して、ファイル ロックを実施します。ただし、すべてのアプリケーションがファイル ロックを利用するわけではありません。たとえばアプリケーション「vi」はファイルをロックせず、代わりにファイル スワップ方式を使用してファイルに対する変更を保存します。

NFSクライアントがロックを要求すると、クライアントはONTAPシステムと通信してロック状態を保存します。ロック状態の保存先は、使用するNFSのバージョンによって異なり、NFSv3ではデータ レイヤに保存され、NFSv4.xではNASプロトコル スタックに保存されます。

NFSv3環境では、NFSプロトコルに付随するNLMプロトコルによってロックが管理されます。そのため、NFSv3でロックが使用されている場合、フェイルオーバー後に古いロックが残っていて、手動でクリーンアップが必要になることがあります。NFSv4.xのロックはリースモデルに基づいて再利用されるため、手動でクリーンアップする必要はありません。NFSファイルロックの詳細については、「ファイルロックの概念」を参照してください。

SVMのファイルロックを表示または削除するには、アドバンスト権限で次のコマンドを実行します。

```
cluster::> set advanced
cluster::*> vserver locks
    break show
```

停止を伴う処理が発生すると、ロック状態が転送されない場合があります。その結果、クライアントからロックが再要求されて、クライアントの新しい場所でロックが再確立されるために、NFSの処理が遅れることがあります。表13は、ロックが保持または失われるシナリオを示しています。

表13）ロック状態のNDO動作

処理	NFSv3の結果	NFSv4.xの結果
ボリューム移動	ロック状態はボリュームとともに移動される。	ロック状態はボリュームとともに移動される。

処理	NFSv3の結果	NFSv4.xの結果
アグリゲートの再配置またはストレージのギブバック処理	ロック状態は移動されず、ロックが使用されている場合は最大45秒間停止します。	ロックの状態は移動されず、ロックが使用されている場合は90秒まで停止します。
LIFの移行（同じノード）	ロック状態はNASプロトコル スタックに保存されない：停止なし。	ロック状態はローカル ノードでそのまま維持される：停止なし。
LIFの移行（別のノード）	ロック状態はNASプロトコル スタックに保存されず、何も移動されない：停止なし。	ロックの状態は移動されず、ロックが使用されている場合は90秒まで停止します。
計画外のテイクオーバー	ロック状態は移動されず、ロックが使用されている場合は最大45秒間停止します。	ロックの状態は移動されず、ロックが使用されている場合は90秒まで停止します。
計画的なテイクオーバー	ロック状態は移動されず、ロックが使用されている場合は最大45秒間停止します。	ロックの状態は移動されず、ロックが使用されている場合は90秒まで停止します。

NFSv4.xロックがフェイルオーバーシナリオに与える影響

フェイルオーバーシナリオ（計画的、計画外、またはテスト）では、NFSv4.xの使用中にクライアントI/Oの一時停止が顕著に発生することがあります。一貫性を維持し、他のクライアントがすでに使用中のファイルをロックできないようにロック状態がネゴシエートされているため、ファイルへの不要な書き込みが発生する可能性があるため、この一時停止は正常です。

このプロトコル固有の一時停止には45～90秒かかることがあり、一部の本番環境では実行できません。この一時停止は猶予期間が原因で発生し、猶予期間がタイムアウトするか、クライアント/サーバ通信が再開するまでNFSロックは維持されます。

詳細については、「NFSv4の猶予期間の仕組み」を参照してください。

ONTAPの猶予期間は、ストレージアーキテクチャの2つの別々の領域でアクティブで、デフォルトで45秒に設定されています。

- **Network（ネットワーク）：**データLIFが（手動またはポート障害によって）別のノードに移動された場合、ロック再生の猶予期間はNFSサーバオプションで設定され `-v4-grace-seconds`ます。この値はNFSサーバレベルで設定されるため、SVMに対するNFSv4.xのすべての処理に影響します。
- **データ/ WAFL：**ストレージフェイルオーバーイベントが発生すると、ロック再生の猶予期間はノードレベルのオプションで設定され `locking.grace_lease_seconds`ます。これはノードレベルで設定されるため、そのノードに対するNFSv4.xのすべての処理に影響します。このオプションは、NFSv4.xを使用したデータアクセスに関与するすべてのノードで設定する必要があります。

ほとんどの場合、NetAppでは、次の理由から猶予期間の値を下げることを推奨していません。

- デフォルト値の45秒は、NFSv4.xクライアントがI/Oを再開する前にロックを再要求する十分な時間を確保できるようにするために役立ちます。現在、すべてのクライアントがRECLAIM_COMPLETEをストレージシステムに送信した場合でも、データレイヤ（ストレージフェイルオーバー用）では、タイマーが切れる前にデフォルトの猶予期間を解除することはできません。この問題に対してバグ1392916がオープンされ、ロック再生のためのフェイルオーバーロジックが強化されています。
- 猶予期間の値が低すぎると、別のクライアントが以前にロックされていたファイルまたはバイト範囲に書き込みを試みる可能性があり、ファイルが破損する可能性があります。
  - 単一の書き込みワークロードの場合（ロックされたファイルに他のクライアントが書き込もうとするリスクがない場合）、猶予期間を短くしても、ある時点で複数のクライアントが1つのファイルに書き込んでいる可能性があるワークロードと同じリスクはありません。
- 再利用にかかる時間は、ロックとクライアントの数によって異なります。ロック/クライアントの数が多い環境では、ロック/クライアントの数が少ない環境よりも再生に時間がかかります。猶予期間の値を安全に設定できるかどうかは、クライアント/ロックの数によって決まります。

ただし、データLIFのフェイルオーバー処理の猶予期間を変更する必要がある場合は、次のコマンドを実行します。

```
cluster::> nfs server modify -vserver DEMO -v4-grace-seconds 45
```

ストレージフェイルオーバー処理の猶予期間を変更するには、次のコマンドを実行します。

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

猶予秒数とリース秒数の差

ONTAP NFSサーバには、NFSv4ロックのタイムアウト動作を制御するオプションが2つあります。それらは似ていますが、同一ではありません。これらのオプションについては、「NFSv4.xロック」セクションで説明しています。

- `v4-lease-seconds` -ONTAPがクライアントにロック/リースを許可する期間を指定します。
- `v4-grace-seconds` -LIFのフェイルオーバーまたは移行中にONTAPとクライアントがロック状態を維持しようとする期間です。

に設定する最小値 `v4-grace-seconds` は `v4-lease- seconds`、値（ONTAPによって適用される）より1秒以上長くする必要があります。これは、リースが45秒間許可されている場合、少なくともその長さの試行を継続し、フェールオーバーイベントがリース秒を超えた場合に備えて追加の時間を確保する必要があるためです。これらのオプションのデフォルト値は次のとおりです。

```
cluster::*> nfs server show -vserver DEMO -fields v4-lease-seconds,v4-grace-seconds
vserver v4-lease-seconds v4-grace-seconds
---------- ---------------------- ----------------------
DEMO     30                45
```

デフォルトでは、NFSv4リースは30秒間クライアントに付与されます。障害イベント（ネットワークの停止やストレージのフェイルオーバーなど）が発生した場合、リースは30秒間存続します。さらに15秒間、ONTAPとクライアントはこれらのロックの再確立を試行します。ネットワークまたはストレージの障害が45秒を超えると、これらのロックは解除され、クライアント/アプリケーションは自動的にロックを再確立する必要があります。

NFSv4.1セッション

ONTAPでは、NFSv4.1セッションがサポートされます。NFSv4.1セッションを使用すると、LIFの移行でNFSv4.1の処理が停止する可能性がありますが、NFSv4.0よりは限定的です。詳細については、RFC-5661のセクション2.10.13を参照してください。NFSv3とNFSv4.xの両方のセッションでパフォーマンスを向上させる方法については、を参照してくださいnconnect。

NFSv4.xでのLIFの移行時の動作

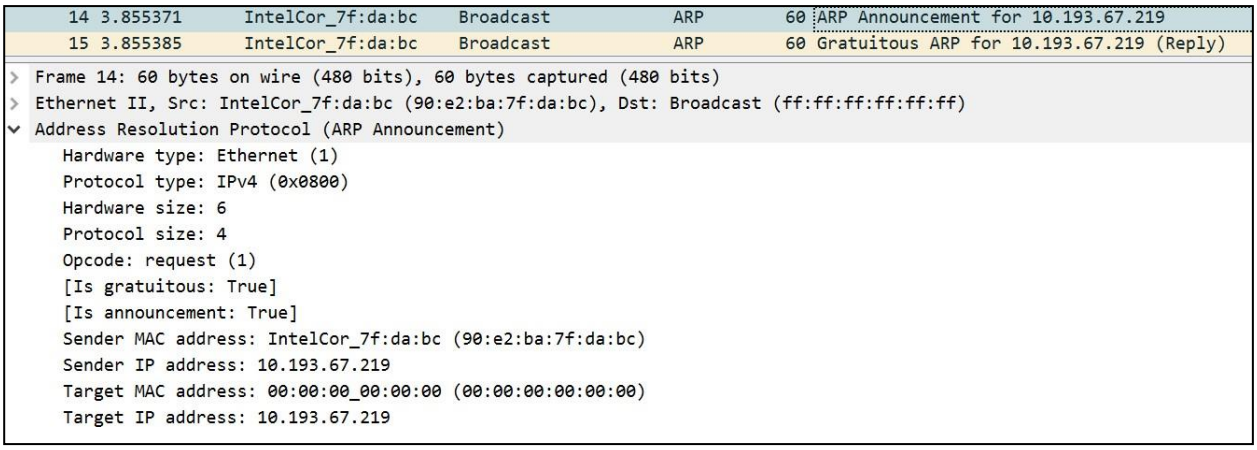
NFSv4.xトラフィックをホストしているデータLIFをONTAPに移行する場合は、LIFを移動しても安全な状態になるまで既存のNFSv4.xトラフィックを休止する必要があります。NFSサーバが安全に移行を許可できると判断されると、LIFが新しい場所に移動され、NFSクライアントによってロック状態が再要求されます。ロック状態の再生は、NFSオプションで制御し `-v4-grace-seconds` ます（デフォルトは45秒）。NFSv4.1セッションでは、ロック状態がNFSv4.1セッションに保存されるため、この猶予期間は不要です。負荷の高いシステム原因では、LIFの移行におけるレイテンシが長くなります。これは、処理が休止されるまでシステムが待機する時間が長くなり、LIFが移行を待機する時間が長くなるためです。ただし、システムが停止するのはロック再要求プロセスの実行中だけです。

NFSv3でのLIFの移行

ストレージフェイルオーバーやポートの障害が原因でLIFを移行すると、ONTAPはそのIPアドレスについてのARP通知をネットワーク経由でブロードキャストし、データLIFのMACアドレスが変更されたことをクライアントに通知します。その結果、クライアントはARPテーブルを更新してその変更を反映します。クライアントがARPエントリを更新できない場合（AppArmorやSELinuxなどのクライアントファイアウォールがARPブロードキャストをブロックしている場合など）、NFSアクセスは引き続き古いMACアドレスを使用しようとします。 そのため、LIFが元のMACアドレスにフェイルバックするか、ARPキャッシュが更新されるまで、アクセ

スエラーやマウントがハングします。図9 は、LIFの移行時のARPを示しています。

図9）LIFの移行時のGratuitous ARP



クライアントのARPキャッシュを表示し、MACエントリをNetAppクラスタのポート情報と比較できます。

たとえば、クライアントには次のような情報が表示されます。

```
# arp -a x.x.x.a
demo.ntap.local (x.x.x.a) at 90:e2:ba:7f:d4:bc [ether] on eno16780032
```

ONTAPには次のような特徴があります。

```
cluster::*> net int show -vserver DEMO -address x.x.x.a -fields curr-port,curr-node
vserver lif   curr-node           curr-port
---------- ------- ---------------------- ------------
DEMO     data2 node2               e2a

cluster::*> net port show -node node2 -port e2a -fields mac
node                 port mac
-------------------- ------ -----------------------
Node2                e2a 90:e2:ba:7f:d4:bc
```

LIFが移行されると、クライアントのARPキャッシュが新しいポートのMACアドレスで更新されます。

```
cluster::*> net int migrate -lif data2 -destination-node node1 -destination-port e2a

cluster::*> net port show -node node1 -port e2a -fields mac
node                 port mac
-------------------- ------ -----------------------
node1                e2a 90:e2:ba:7f:da:bc

# arp -a x.x.x.a
demo.ntap.local (x.x.x.a) at 90:e2:ba:7f:da:bc [ether] on eno16780032
```

NFSで使用中のデータLIFの安全な運用停止

データSVMのIPアドレスの運用を停止する場合があります。ただし、これらのIPアドレスがNFSマウント/SMB共有で頻繁に使用されていると、環境が意図せず停止する可能性があります。これらのIPアドレスがDNSロードバランシングゾーンの一部である場合、これはより問題になる可能性があります。削除されたIPアドレスへのアクティブな接続がクライアントで試行される可能性があります。すでにそれらのIPアドレスに接続されていたクライアントでは、SVMから削除されていても、そのIPアドレスへの処理の送信が試行されます。TCP接続では、再マウントが実行されるまでIPアドレスは自動的に更新されません。

データLIFの運用停止、削除、または管理ステータスのdownへの変更を計画している場合は、次のガイドラインを使用してください。

- データLIFがDNSロードバランシングゾーンまたはラウンドロビンに参加している場合は、メンテナンス時間のしばらく前に、それらのLIFをゾーンから削除してください。ONTAP内蔵DNSの場合は、を使用し `net int modify -dns-zone none`ます。オフボックスDNSの場合は、これらのIPアドレスのA/AAAAレコードをすべて削除します。目標は、新しいクライアントがIPアドレスにマウントされないようにすることです。
- インターフェイスをに変更する前に、 `-status-admin down`既存のネットワーク接続を確認します。
  - ONTAP 9.6以前では、を使用し `network connections active show`ます。
  - SMB / CIFSクライアントの場合は、を使用し `cifs session show`ます。
  - ONTAP 9.6以降のNFSクライアントにはを使用し `nfs connected-clients show`ます。
- マウントが接続されている非アクティブなクライアントが存在する可能性があることに注意してください。これらのリストには、最近アクセスされた接続（約24時間程度）のみが表示されます。ただし、これらのクライアントはマウントを使用しないため、インターフェイスの運用を停止するときに問題ではない可能性があります。確認するには、数がゼロに達するまで数日間接続を確認してください。または、メンテナンス時間についてクライアントに通知しておく必要があります。
- 運用停止するIPアドレスにミッションクリティカルなクライアントがアクセスしていないことを確認したら、インターフェイスを削除できます。複数のインターフェイスを削除する場合は、一度にすべてのインターフェイスを削除しないでください。これにより、ミスが発生した場合のシステム停止を最小限に抑えることができます。
- network connections active showコマンドの例については、「クラスタ内のアクティブなNFS接続の表示」を参照してください。
- nfs connected-clients showの例については、「nfs connected-clients の出力例」を参照してください。

pNFSとLIFの停止

pNFSを使用すると、NFSv4.1クライアントは、読み取りまたは書き込み中のデータが格納されているボリュームを所有するノードにトラフィックをリダイレクトできます。これにより、データが局所的に格納されるため、パフォーマンスが向上します。そのためには、データアクセスに関与するクラスタの各ノード上のSVMにデータLIFがあり、常に可能なローカルパスが確保されている必要があります。SVMにルーティングされないLIFがあると、pNFSによってそれらのデータLIFが選択され、システムが停止する可能性があります。pNFSを使用する場合は、NFSv4.1クライアントと通信できないSVMにデータLIFを追加しないでください。

データLIFがオフラインになると、データI/Oはマウントが接続されたIPアドレスであるMDSにフォールバックされます。データI/Oの実行中にデータLIFが停止した場合は、クライアントがMDS接続にフォールバックしてデータI/Oを再開するまでに最大60秒の遅延が発生します。

WDELAY/No_WDELAY

一部のアプリケーション（RedHat OpenShiftなど）では、NFSサーバエクスポートオプション NO_WDELAY に固有の要件があります。これは、書き込みが保証されていない一部のNFSサーバがアプリケーションの相互運用性で持つ可能性がある、キャッシュの不一致、パフォーマンス、およびその他の問題から保護するためのものです。ONTAPではこのエクスポートオプションは使用されません。ONTAPのすべての書き込みは、書き込みがNFSクライアントから確認応答されたあとすぐに使用可能になるためです。

直接接続型NFS

直接接続NFSとは、NFSクライアントからストレージシステムへの直接接続のことです。NFSはイーサネットベースのプロトコルであるため、技術的な理由はありません。ただし、NetAppクラスタでは、NFS接続に複数のノードが使用されます。ストレージフェイルオーバー、ケーブル障害、ポート障害などが発生した場合、直接接続されているクライアントはNFSエクスポートと通信できません。

最高の信頼性とパフォーマンスを得るためには、本格的なネットワーク経由でNFSを使用することを強く推奨します。

ボリューム形式の選択：FlexGroupかFlexVolか

NFSワークロードで使用するボリュームを導入する場合は、次の2つのボリューム形式を選択できます。

- **FlexVolボリューム** は、ONTAPで使用できる標準のボリュームタイプで、単一ノードのハードウェアにまたがるボリュームです。
- **FlexGroup** は、クラスタ内の複数のハードウェアドメインにまたがる複数のFlexVolメンバーボリュームで構成されるボリュームです。これには、FlexVolよりも次のような多くの利点があります。
  - ボリュームサイズが100TBを超える（テスト済みの20PB）。
  - ファイル数が20億を超える（テスト済みの4、000億）。
  - 取り込み負荷の高いワークロードで2~6倍のパフォーマンスを提供するマルチスレッドメタデータ処理。
  - クラスタ内の複数のノードを使用して、ワークロードを自動的に分散する機能。
  - 使いやすいFlexVolに似た管理機能。
  - ボリュームの容量が上限に達したときに無停止で拡張できます。

ほとんどのNFSワークロードでは、FlexGroupボリュームはFlexVolボリュームよりも多くのメリットをもたらします。この決定を行う際の主な注意点は、ボリューム形式間の機能のパリティを確認して、ご使用の環境で必要な機能がサポートされているかどうかを確認することです。導入や決定ポイントの詳細など、FlexGroupボリュームの詳細については、TR-4571：『NetApp FlexGroup Volume Best Practicesand Implementation』を参照してください。

NFS監査

次のセクションでは、NFS監査の設定と使用について説明します。NFS監査では、NFSv4.x監査ACE（UNIXセキュリティ形式）またはWindows監査ACE（NTFSセキュリティ形式）のいずれかを使用できます。

NFSカンサセツトアツフ

NFS監査を設定するための主な要件は、監査を必要とするボリュームに監査ACEを設定することです。監査ACEにはWindowsまたはNFSv4.xを使用できますが、NFSのみの環境ではNFSv4.xを使用する必要があります。そのため、NFSサーバでNFSv4.xを有効にし、NFSv4.x管理クライアントを使用して監査を設定する必要があります。

ONTAPシステムテノカンサノユウコウカ

NFSv4.x ACLの詳細については、「NFSv4.x ACL」を参照してください。

NFSv4.xおよびNFSv4.x ACLを有効にしたら、次のコマンドを使用してNFS監査を有効にします。

```
cluster::> vserver audit create -vserver nfs -destination /unix -rotate-size 100MB
```

このコマンドは `/unix` 、という名前のSVMのジャンクションパスでのNFSアクセスとCIFSアクセスの監査を許可します `nfs`。ONTAPシステムで監査を有効にしたら、監査ACEを作成する必要があります。

注: 継承可能な監査ACEを使用する場合は、アクセスの問題を回避するために、継承可能なALLOW ACEまたはDENY ACEを親ディレクトリに少なくとも1つ作成してください。詳細については、バグ959337を参照してください。

NFSv4監査ACEの作成

NFSv4監査ACEを作成するには、監査を有効にしたボリュームをNFSv4.xを使用してマウントします。ボリュームがマウントされたら、監査が必要なボリューム、ファイル、ディレクトリで監査ACEを作成します。

監査ACEを使用して、以下のようなさまざまな処理に対するALLOWまたはDENYを追跡することができます。

- 読み取り
- 書き込み
- Execute
- Append
- 削除

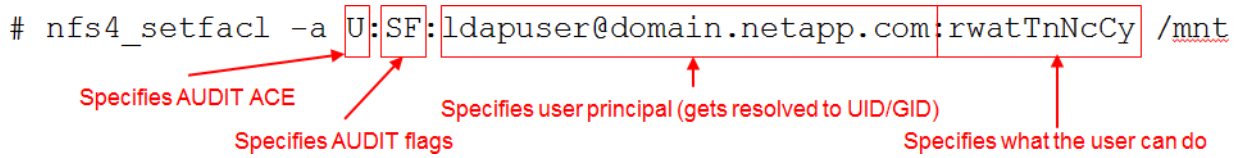
NFSv4のすべてのACE権限については、http://linux.die.net/man/5/nfs4_aclを参照してください。

各Linuxクライアントでは、NFSv4.x ACEの割り当て方法が異なります。RHEL / CentOS / Fedoraでは、コマンド `nfs4_setacl` とが `nfs4_getacl` 使用されます。

監査ACEでは、フラグを使用して、監査の対象を成功、失敗、またはその両方のいずれにするかを指定します。監査ACEでは、ACEタイプUを使用します。

図10 は、NFSv4監査ACEの設定例を示しています。

図10）NFSv4監査ACEの設定例



監査ACEを適用すると、監査対象のユーザがアクセスを試み、ボリューム上のXMLファイルにイベントが記録されます。

ログに記録されたNFS監査イベントの例については、「NFS監査イベントの例」を参照してください。

NFSノヘストフラクテイス

次のセクションでは、NFS環境のベストプラクティスについて説明します。ベストプラクティスは、NFSを使用して導入する際に考慮すべき推奨事項にすぎません。ベストプラクティスは難しい要件ではなく、多くの場合、環境内のさまざまな要因に依存しています。

ONTAPの一般的なベストプラクティス

次に、ONTAPの一般的なベストプラクティスを示します。

- ONTAPの最新バージョンにアップグレードして、バグ修正と最新のNFS機能を確認してください。
- Active IQを使用して、アップグレードの推奨事項とプロアクティブな修復を行います。
- AutoSupportを有効にすることで、シームレスで効率的なサポートを実現できます。
- NFSv3環境でvsrootボリュームの負荷共有ミラーを設定します。
- パフォーマンスを最大限に高めるには、NetAppオールフラッシュFASシステムを使用してください。
- すべてのStorage Efficiencyを有効にして、ストレージ利用率を最大限に高めます。
- アグリゲートとストレージの構成のベストプラクティスについては、製品ドキュメントを参照してください。
- ほとんどの場合、特に設定を変更する理由がないかぎり、デフォルトのままにしておいてください。
- Active IQ Unified Managerをインストールし、ONTAPクラスタを監視するようにを設定します。
- ストレージフェイルオーバー、ボリューム容量アラートなどのイベントに対してプロアクティブなアラートを設定します。
- 特にファイル数の多い環境では、NFSワークロードにFlexGroupボリュームを導入することを検討してください。

NFS環境でのONTAPデータLIFのベストプラクティス

ONTAPを使用すると、ストレージ管理者は次のメリットを得ることができます。

- シームレスなスケールアウト ストレージ
- マルチプロトコル ユニファイド アクセス（NFS、CIFS、SAN）
- ノンストップ オペレーション

これらを実現するのが、SVMを使用したセキュアなマルチテナント アーキテクチャです。

SVMとは

Storage Virtual Machine（SVM）は論理的なストレージコンテナで、フレキシブルボリューム、論理インターフェイス（LIF）、エクスポート、CIFS共有などのストレージリソースを所有します。クラスタ内のストレージブレードセンターと考えてください。クラスタ内の物理ハードウェア リソース（ネットワーク ポート / VLAN、物理ディスクを含んだアグリゲート、CPU、RAM、スイッチなど）を他のSVMと共有します。データアクセスは、クラスタ内の場所に関係なく、SVMが所有する任意のデータネットワークインターフェイスで実行できます。負荷をクラスタ全体に分散することができるため、最大限のパフォーマンスと効率を実現したり、SaaS機能を提供できるといったメリットがあります。また、単一のSVMを使用して、モノリシックストレージデバイスを環境に提供することもできます。

データLIFに関する考慮事項

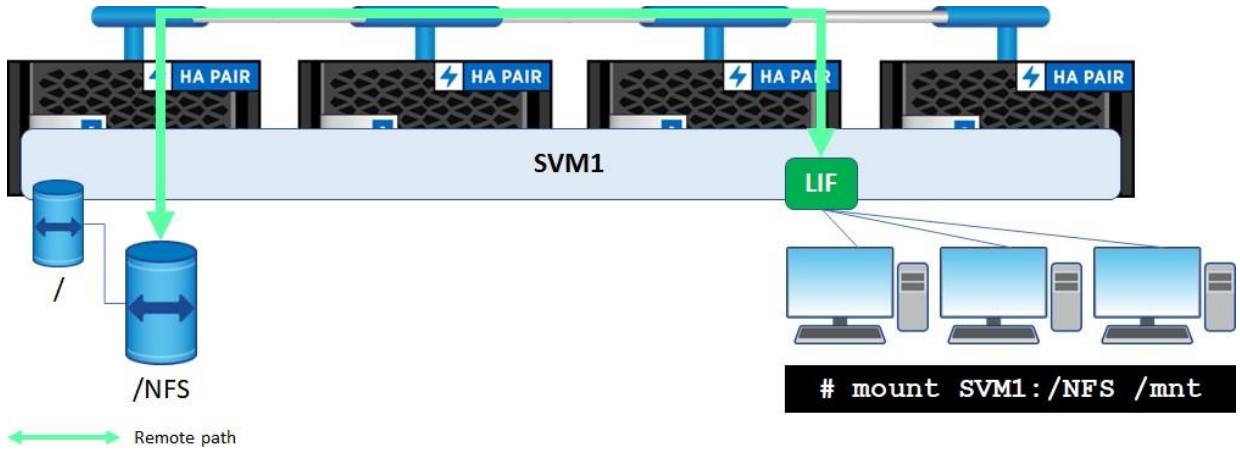
データLIFは、有効なブロードキャスト ドメインに追加された、クラスタ内の任意のポートに配置できます。データLIFはSVM対応のルーティング メカニズムで設定されるため、有効なデータLIFがクラスタ内のどこにあるかにかかわらず、SVM内のイーサネット トラフィックは正しく転送されます。NASインタラクション用のネットワークを設計する場合は、2つのアプローチのいずれかを実行できます。

オプション1：簡易化アプローチ–SVMあたりのLIFは1つ

基本的に、ONTAP内のNASデータにアクセスするために必要なのは、ネットワーククライアントにルーティング可能な単一のネットワークIPアドレスだけです。多くの環境では、1つのネットワークインターフェイスでNASワークロードを処理できます。基盤となる物理ネットワークポートに障害が発生した場合、またはストレージノードがHAパートナーにテイクオーバーされた場合、ネットワークIPアドレスはクラスタ内の別の動作中ポートに移行されます。単一のネットワークインターフェイスを使用すると、必要なIPアドレスの数が削減されますが、ワークロードで使用できる可能性のあるネットワーク帯域幅も制限されます。すべてのNASトラフィックをクラスタ内の1つのノードに送信すると、使用可能なリソース（CPUやRAMなど）の数も制限されるため、高いスループットが求められるワークロードや、数百から数千のクライアントを接続するワークロードの場合は、オプション2を選択することをお勧めします。

図11 は、単一のLIFのNASの連携を示しています。

図11）単一LIFのNASの連携



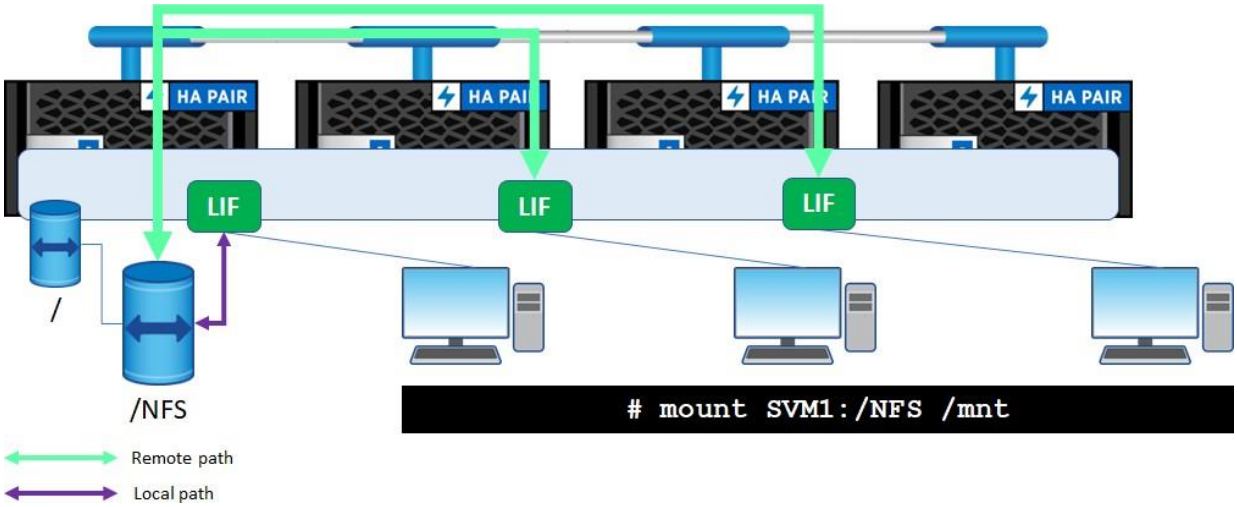
オプション2：パフォーマンスアプローチ–SVMごとに複数のデータLIFを使用

ONTAPクラスタでは、複数のノードをNAS接続およびストレージに使用できます。NASを使用するONTAPクラスタは、最大24ノードまでしかスケールできません。複数のノードとは、CPU / RAM /ネットワークインターフェイスなど、複数の物理リソースを意味します。そのため、1つのSVMに複数のデータLIFがあると、NASワークロードのパフォーマンスが大幅に向上する可能性があります。ノード間でネットワーク接続を分散すると、CPUとネットワークポートの競合が軽減され、ノードのTCP接続数が多すぎるシナリオが回避されます。NAS接続のネットワークロードバランシングには、ラウンドロビンDNS、内蔵DNS、または標準のロードバランシングハードウェアを利用できます。内蔵DNSとその設定方法の詳細については、TR-4523：『ONTAPにおけるDNSロードバランシング』を参照してください。

可能なかぎり最高のパフォーマンスが必要な場合や、多数のクライアントが1つのNASデバイスに同時にアクセスする場合は、SVMごとに複数のデータLIFを作成するのが適切なアプローチです。さらに、NFSリファーラル、CIFSオートロケーション、pNFSなどのNAS機能のロードバランシングを使用すると、データが存在する各ノードにデータLIFが必要になります。

図12 は、複数のLIFのNASの連携を示しています。

図12）複数のLIFのNASの連携



データLIFの局所性に関する推奨事項

ONTAPでは、ボリュームがクラスタ内でどこにあるかに関係なく、NFSリファーラル、CIFSオートロケーション、pNFSなどのデータ局所性機能をNASトラフィックに利用できます。NFSリファーラルおよびCIFSオートロケーションの場合、最初のTCP接続は、要求されたボリュームに対してローカルなネットワークインターフェイスに自動的にリダイレクトされます。使用しているボリュームがFlexGroupボリュームの場合は、NFSリファーラルとCIFSオートロケーションを使用しないでください。

pNFSは最初のマウント要求でメタデータパスを提供しますが、すべての読み取りと書き込みは、pNFSレイアウト呼び出しによって自動的にローカルボリュームにリダイレクトされます。pNFSは、NFSv4.1プロトコルでのみ使用でき、pNFSをサポートするNFSクライアントでのみ使用できます。FlexGroupボリュームでpNFSを使用できるのはONTAP 9.7以降のみです。pNFSの詳細については、「パラレルネットワークファイルシステム」を参照してください。

オートロケーション機能を使用しない場合、データLIFの局所性を管理してクラスタネットワークを回避すると、管理が複雑になりますが、ほとんどのNFSワークロードのパフォーマンスへの影響はほとんどありません。NAS接続はボリュームに対してローカルなデータLIFに接続するのが理想的ですが、FlexGroupボリューム/スケールアウトNASや大規模なクラスタバックエンドネットワークでは、これはそれほど重要ではありません。

データローカリティのメリットと考慮事項

次のセクションでは、ONTAPでのデータローカリティのメリットと考慮事項、およびこれらの概念へのシンプルさを念頭に置いたアプローチ方法について説明します。

ノード間で負荷を分散し、クラスタ内の使用可能なすべてのハードウェアを活用

ボリュームとネットワークインターフェイスを作成する場合は、パフォーマンスヘッドルームを最大化するために、クラスタ内の複数のノードにワークロードを導入することを検討してください。使用しないハードウェアはコストの無駄です。

シンプルなアプローチ： ONTAPでは、ONTAP System Managerを使用すると、ストレージのプロビジョニングが自動化されます。これには、使用可能なパフォーマンスヘッドルームが考慮され、利用率の低いノードに新しいボリュームが配置されます。さらに。FlexGroupボリュームは、クラスタ内の複数のノードにプロビジョニングし、単一のネームスペースにワークロードを自動的に分散します。

複数のクラスタ ノード間でネットワーク接続を分散できる

クラスタはSVM同様に単一のエンティティです。ただし、その基盤となるハードウェアにはそれぞれ上限があります（接続数など）。

シンプルなアプローチ： SVMごとに複数のデータLIFを作成し、ONTAPの内蔵DNS機能を利用して、それらのインターフェイスをDNSラウンドロビン名またはDNS転送ゾーンの背後にマスクします。さらに、FlexGroupボリュームを活用して、複数のノードにワークロードを分散します。

ボリューム移動の際にデータ局所性を実現できる

ボリュームを別のノードに移動する場合に、すべてのノードにSVMのデータLIFがあればデータへのローカルパスを確保できます。ONTAPのボリュームを新しいノードに移動する場合、NASクライアントでは既存のTCP接続が維持されます。その結果、これらのNAS処理はクラスタネットワークを経由します。

シンプルなアプローチ： 何もしない。ほとんどの場合、NASクライアントはこれらのNAS共有のパフォーマンスに違いを認識しません。NFSv4.1を使用している場合は、pNFSの使用を検討してください。

多数のNFSクライアントでネットワークポートが枯渇している

多数のクライアントがNFS経由で接続されている環境では、マウント ポートとNFSポートの数がデフォルトで1,024に制限されていることを念頭に置く必要があります。

この番号は、次のオプションで制御されます。

```
mount-rootonly
nfs-rootonly
```

デフォルトでは `mount-rootonly`、はEnabledに設定され、`nfs-rootonly` Disabledに設定されています。

マウントやNFS処理に使用されるポートの数が足りなくなり、ポートが使用可能になるまで後続のマウントやNFS処理がハングする場合があります。

環境内に数千ものクライアントがNFS経由でマウントされ、I/Oを生成している場合、NFSサーバ上のすべてのポートが使い果たされる可能性があります。たとえば、ESXでNFSデータストアを使用するシナリオが1つあります。これは、一部の従来のベストプラクティスではデータストアごとにデータLIF / IPアドレスが必要になるためです。このシナリオでは、多数のボリュームやデータストアがある環境でNFSポートがオーバーランする状況が発生しました。この場合は `mount-rootonly nfs-rootonly` 、NFSサーバでオプションやオプション（あるいはその両方）を無効にして対処します。この解決策により、1～1、024個のポート範囲の制限が解除され、NFSサーバで最大65、534個のポートを使用できるようになります。これらのオプションの詳細については、「rootonlyオプション-nfsrootonlyおよびmountrootonly」を参照してください。

注： ONTAPでのESX / NFSのベストプラクティスについては、TR-4597：『VMware vSphere with ONTAP』を参照してください。

この状況は、送信元ポート（クライアント側）だけに影響します。サーバ側マウントポート、ポートマッパーポート、NFSポート、およびNLMポートは、ONTAPによって指定されます。

ネットワークアドレス変換でのNFS

NFSは、特定の処理が完了したことを確認するために、応答キャッシュを保持して特定の処理を追跡します。このキャッシュは、送信元ポートと送信元IPアドレスに基づいています。NFSの処理でネットワークアドレス変換（NAT）が使用されている場合、送信元のIPまたはポートが転送中に変更され、データの耐障害性の問題につながる可能性があります。NATを使用する場合は、データの整合性を維持するために、NFSサーバのIPとポートの静的なエントリを追加する必要があります。

また、NATがアイドルセッションを処理する方法が原因でNFSマウントがハングする問題も発生する可能性があります。NATを使用する場合は、問題を回避するために、アイドルセッションを考慮し、無期限に開いたままにしておく必要があります。NATでは、NLMロックの再利用に関する問題も発生する可能性があります。

最終的に、NFSを使用するNATのベストプラクティスは、可能であればNATの使用を避け、SVMにデータLIFを作成することです。NATが必要な場合は、NATベンダーと協力して、NFSが動作するようにNATを適切に設定してください。

LIFのサービスポリシー

ONTAP 9.6以降では、LIFのサービスポリシーが導入されています。これは、ONTAPのネットワークデータインターフェイスのロールの概念に代わるものです。LIFポリシーをネットワークインターフェイスに適用または削除すると、ネットワークインターフェイスを再作成しなくてもトラフィックを許可または禁止できます。

次のコマンドを実行すると、インターフェイスに設定されているサービスポリシーを確認できます。

```
cluster::*> net int show -vserver DEMO -lif data -fields service-policy
  (network interface show)
vserver lif service-policy
---------- ------ ----------------------
DEMO    data default-data-files
```

LIFのサービスポリシーでは複数のデフォルトポリシーが作成されますが、カスタムポリシーを追加することもできます。これらは、SAN、NAS、または管理トラフィックを許可するデフォルトのポリシーです。1つのデータLIFに同時に割り当てることができるポリシーは1つだけです。

```
cluster::*> network interface service-policy show -vserver DEMO
Vserver   Policy                       Service: Allowed Addresses
--------- ---------------------------------- ----------------------------------------------------
DEMO
          default-data-blocks          data-core: 0.0.0.0/0, ::/0
                                       data-iscsi: 0.0.0.0/0, ::/0
                                       data-fpolicy-client: 0.0.0.0/0, ::/0

          default-data-files           data-core: 0.0.0.0/0, ::/0
                                       data-nfs: 0.0.0.0/0, ::/0
                                       data-cifs: 0.0.0.0/0, ::/0
                                       data-flexcache: 0.0.0.0/0, ::/0
                                       data-fpolicy-client: 0.0.0.0/0, ::/0

          default-management           data-core: 0.0.0.0/0, ::/0
                                       management-ssh: 0.0.0.0/0, ::/0
                                       management-https: 0.0.0.0/0, ::/0
                                       data-fpolicy-client: 0.0.0.0/0, ::/0
```

NFSのみまたはCIFS / SMBのみを許可するポリシーを作成する場合は、 `network interface service-policy createnetwork interface service- policy add-service` またはを使用してサービスを追加または削除できます `network interface service-policy remove-service`。これらはすべて、システムを停止することなく実行できます。

詳細については、ONTAP 9.6以降のLIFとサービスポリシーを参照してください。

NFSセキュリティのベストプラクティス

NFSはNetwork File System（ネットワークファイルシステム）の略ですが、これまではSecurity（セキュリティ）の略ではなくとも呼ばれてきました。NFSv3以前では、クライアントIPアドレス、数値ユーザ、グループID、およびその他の識別情報がクリアテキストでネットワーク経由で渡されるため、セキュリティの範囲が制限されていました。

NFSv3セキュリティのデフォルトのセキュリティメカニズムは、エクスポートルールと基本的なユーザモードビット権限であるため、アクセス用のIPアドレスとユーザIDのスプーフィングは非常にシンプルで簡単です。そのため、NFS環境のセキュリティを強化するために、次の推奨事項を参考にしてください。

NFSバージョンの選択

NFS環境を導入する際には、使用しているNFSバージョンを考慮する必要があります。NFSv3が圧倒的に普及しているNFSバージョンですが、NFSv4.xの人気は高まっており、より厳格なセキュリティ要件を求める企業が増えています。

- セキュリティが環境で最も重要な要素である場合は、NFSv4.xはセキュリティを考慮して設計されているため、少なくともテストする必要があります。
- NFSv3は、物理パフォーマンスを優先する場合は、NFSv4.xよりもステートレスで優れたパフォーマンスが得られるため、慎重に選択してください。
- ロックが重要な場合は、NFSv4.xが最も堅牢で耐障害性に優れたオプションを提供します。
- 複数の機能を組み合わせる必要がある場合は、ワークロードに最も適したNFSバージョンをテストすることを推奨します。

「NFSのバージョンに関する考慮事項」セクションでは、NFSのすべてのバージョン、利点、機能、および移行に関する考慮事項について詳しく説明します。

NFSv3ノホコ

NFSv4.xはセキュリティ上NFSv3よりも優れていますが、プロトコルのセキュリティを強化するためにはいくつかの手順を踏むことができます。

エクスポート ポリシー ルールに関する考慮事項

エクスポートポリシーとルールは、NFSエクスポートを保護するための最初のゲートウェイです。これらのルールを使用すると、1つ以上のクライアントを指定し、エクスポートをマウントするときに受け取るアクセスレベルを定義できます。これらのルールは、ファイルレベルの権限よりも優先されます。たとえば、「read/write」（rwrule）をneverに設定すると、UNIX権限が777であっても、そのクライアントはエクスポートに書き込むことができません。NFSエクスポートポリシーおよびルールの詳細については、「エクスポートの概念」を参照してください。

エクスポートポリシーとエクスポートルールの設定方法によって、クライアントにエクスポートのマウントを許可するかどうかが決まります。NFSv3エクスポートを保護する場合は、次の点を考慮してください。

- エクスポートポリシーにルールが設定されておらず、ボリュームまたはqtreeに適用されている場合は、どのユーザもマウントへのアクセスを許可されません。
- エクスポートポリシールールに特定のクライアント（IP、ホスト、名前、ネットグループ/、またはサブネットを使用）が含まれていない場合は、 そのクライアントはマウントできません。
- vsrootボリュームのエクスポートポリシールールで特定のクライアントへの読み取りアクセスが許可されていない場合、そのクライアントはエクスポートをマウントできません。エクスポートポリシールールでは、クライアントがボリュームへのパスをトラバースできるように読み取りアクセスが必要です。Vsrootはエクスポートパスのにあります。
  - Vsrootは、作成時にデフォルトのエクスポートポリシーを使用します。デフォルトでは、「デフォルトポリシー」にはエクスポートルールはありません。
- vsrootのエクスポートポリシールールに対してclientmatchを設定してパスのトラバーサルを許可する場合は、ネットグループ、サブネット、またはクライアントのリスト別に設定することを検討してください。エントリを0.0.0.0/0または0/0に設定しないでください。これにより、すべてのクライアントがエクスポートをトラバースまたは読み取りできるようになります。詳細については、「vsrootへのアクセス制御」を参照してください。

- クライアントにルートアクセスを許可（または禁止）するには、[スーパーユーザ]エクスポートポリシールールオプションを使用します。詳細については、「すべてのUIDを単一のUIDにマッピングする（squash_all）」を参照してください。
- マルチプロトコルNAS環境（CIFS / SMBおよびNFSアクセス）では、NTFSセキュリティを使用するボリュームのNFSマウントでの権限または所有者の変更に対するONTAPの応答方法は、エクスポートポリシーの設定によって異なります。オプションは、[失敗（エラーを送信）]または[無視（エラーなしでサイレントに失敗）]です。
- を設定する `-rw-rule-ro-rule` か `-superuser` [None]に設定すると、で設定したユーザへのクライアントのアクセス権が引き下げられます `-anon`。ほとんどの場合、は `-anon` 65534に設定する必要があります。`-anon` 管理ホストへのrootアクセスを許可するには、0に設定するか、特定のクライアント/クライアントセットのファイル所有権を制御するために明示的なユーザに設定することは例外です。
- エクスポートclientmatchでホスト名とIPアドレスを設定するのではなく、ネットグループを使用して `-clientmatch` フィールドのホスト/クライアントリストをマスクすることを検討してください。ネットグループは、SVM上のローカルファイル、LDAP、またはNISに配置できます。
- 読み取り/書き込みアクセスを実行しているクライアントの場合、`-superuser -anon` ルートアクセスを無効にするには、Noneに設定し、65534に設定します。
- 管理/ルートレベルのタスク用にクライアントのグループを設定し、ルートアクセスを許可するようにエクスポートポリシーを設定することを検討してください。rootアクセスを許可する方法の詳細については、「rootユーザの制御例」を参照してください。
- NFSv3のマウントにKerberosを使用する場合は `sys`、との両方を `krb5*` ルールに含めてください。NFSv3は補助プロトコルを使用し、NFS部分にはKerberosのみを使用します。NFSv3ではエクスポートポリシールールへのアクセスをに制限する `krb5*` と、`sys` ポートマッパーやマウントなどにアクセスが必要になるため、マウントが失敗します。
- `any` では `-rw-rule`、`-ro-rule` `superuser`を使用しないでください。許可すると `any`、ポリシールールのセキュリティ効果が低下します。代わりに、これらのオプションで必要な認証方法を指定します。
- `any` ではを使用しないで `-protocol`ください。使用しないプロトコルにアクセスできるようになるためです。両方のNFSv3でマウントにNFSv4.xアクセスする必要がある場合は、プロトコルをに設定します `nfs`。マウントへのアクセスをNFSv3のみ、またはNFSv4.xのみに許可する場合は `nfs3` 、またはを指定します `nfs4`。

クライアントファイアウォール

デフォルトでは、NFSクライアントはファイアウォールを有効にします（RHELのSELinuxやUbuntuのAppArmorなど）。場合によっては、これらのファイアウォールがNFSの処理に影響を与える可能性があります。NFSアクセスの問題が発生した場合は、必ずクライアントのファイアウォールルールを確認し、トラブルシューティング中に必要に応じてファイアウォールサービスを停止します。

ファイアウォールポートルール

NFSv3では、すべてのNFSv3処理へのフルアクセスを提供するために、ファイアウォールで許可する必要があるポートがいくつかあります。一般に、ファイアウォールには、既知のNFSポートのリストを保持するNFSなどのルールが組み込まれています。ただし、多くのセキュリティ環境では、既知のポートを変更してセキュリティを強化することを推奨しています。「ONTAPのデフォルトのNFSポート」に記載されているポートは、ONTAPがNFSv3にデフォルトで使用しているポート、変更可能なポート、適用先のNFSバージョンを示しています。

次に、ファイアウォールルールの一般的な推奨事項とガイダンスを示します。これらはNetAppの要件ではなく、NFS環境のセキュリティを確保するための単なるアイデアです。

- NFS補助プロトコルのデフォルトのポート番号をwell-knownでないポートに変更し、必要に応じてファイアウォールポートを調整することを検討してください。
- 環境で必要なNFSv3ポートだけを許可することを検討してください。たとえば、rquotaが使用されていない場合は、ファイアウォールで許可しないでください。ただし、NFSマウントには複数のポートが必要なため、これらのポート番号は常に許可しておく必要があります。
  - NFSアクセスを許可するには、portmapper（111）、mountd（635）、およびnfs（2049）が必要です。その他の補助ポート（NLM、NSM、rquota）は、アプリケーションまたはクライアントで必要な場合にのみ必要です。必要に応じて環境でテストします。

- NFSマウントでは、クライアントにソースポートが生成されます。デフォルトでは、ONTAPは許可される送信元ポートの範囲を1～1024()の範囲に設定します-mount-rootonly。場合によっては、この範囲のポート番号が、マウントするクライアントの数に十分対応できないことがあります。送信元ポート番号がさらに必要な場合は、-mount-rootonly [Disabled]に設定し、その変更に対応するようにファイアウォールルールを変更します。詳細については、「rootonlyオプション-nfsrootonlyおよびmountrootonly」を参照してください。
- NFS処理では、を使用して制御できるさまざまなソースポートも使用され -nfs-rootonlyます。デフォルトでは、この値はDisabled（無効）に設定されています。つまり、ポート範囲は1～65536です。使用するNFSクライアントの数が少ない（100以下）場合は、このオプションをenabledに設定して、ポート範囲を1 ~ 1024に減らしてセキュリティを強化することを検討してください。
- ローカルネットワークの外部でNFSポートを開かないようにします。WAN経由のNFSが必要な場合は、エンドツーエンドの暗号化にNFS Kerberosとkrb5pの使用を強く検討してください。または、NetApp FlexCacheボリュームを使用してリモートサイトへのNFSトラフィックをローカライズすることを検討してください。NetApp FlexCacheボリュームでは、TLS 1.2を使用して通信が暗号化されます。FlexCacheボリュームの詳細については、TR-4743：『FlexCache in ONTAP』を参照してください。
- UNIX ID管理のエクスポートポリシールールとネームサービスは、外部ネームサーバ（DNS、LDAP、NIS、Active Directoryなど）に依存する場合があります。これらのネームサーバのトラフィックがファイアウォールルールで許可されていることを確認してください。
- 一部のファイアウォールでは、一定の時間が経過するとアイドル状態のTCP接続がドロップされることがあります。たとえば、クライアントでNFSマウントが接続されていて、しばらく使用していない場合、アイドル状態とみなされます。この場合、ネットワーク接続がファイアウォールによって切断されているため、マウントへのクライアントアクセスが停止する可能性があります。キープアライブはこれを防ぐのに役立ちますが、古いセッションからのパケットをアクティブに拒否するようにファイアウォールを設定するか、ONTAP NFSサーバオプション -idle- connection-timeout とを設定して、この -allow-idle-connection問題に対処することを推奨します。これらのオプションはONTAP 9.5で導入されました。詳細については、バグ1072144を参照してください。

権限

エクスポートポリシーではクライアントの読み取り権限と書き込み権限を設定できますが、エクスポートルールだけで特定のユーザとグループに権限を設定することはできません。また、エクスポートポリシーを使用した場合、読み取りや書き込みよりも詳細な処理を実行することはできません。そのため、ファイルとフォルダの権限が必要になります。NFSv3では、デフォルトで、ファイルおよびフォルダに対してモードビットを使用して基本的な権限モデルが使用されます。つまり、権限はファイル/フォルダの所有者とそれ以外のすべてのユーザに対してのみ明示的に制御できます。さらに、権限のレベルは RFC-1813で定義されているものに制限されます。

図14 は、各権限モードビットが0～7の範囲で何を行うかを示しています。

表14）UNIXモードビットレベル

記号表記	モードビット値	Access Level（アクセス レベル）
-	0	アクセス不可
rwx	7	すべてのアクセス
RW -	6	読み取り/書き込み
r-x	5	読み取り / 実行
R --	4	読み取り
- WX	3	書き込み / 実行
-w-	2	書き込み
-- x	1	Execute

NFSv4.x管理クライアントを使用してファイルシステムをマウントし、NFSv4.x ACLを追加することもできます。追加したACLはNFSv3クライアントで維持されます。これにより、NFSv3環境に対してより詳細な権限が提供されます。

NFSv3のセキュリティ強化に役立つその他の推奨事項を次に示します。これらは一般的な推奨事項であり、すべての環境に適用されるわけではないため、必ずご使用の環境でテストしてください。

- vsrootボリュームについては、所有者とグループを[Read and Execute]に、それ以外のユーザは[Execute Only]の権限（551）に制限します。Vsrootは一般的に小さいので、可能な限りユーザーがVsrootに書き込めないようにしたいと思います。所有者とグループ以外のすべてのユーザを実行権限に制限すると、パスのトラバーサルのみが許可され、vsrootボリュームにファイルやディレクトリを表示することはできません。
- ボリュームの所有者またはグループへのルートアクセスを許可する場合を除き、ボリューム/ qtreeの権限を7に設定しないでください。
- NFSファイルシステムへのアクセスを管理する場合は、UNIX権限の標準的なベストプラクティスに従います。
- 権限管理を強化するには、NFSv4.x ACLまたはNTFSセキュリティ形式のボリューム権限（マルチプロトコルNASを使用している場合）の使用を検討してください。
- UNIX IDにLDAPまたはNISを使用している場合は、ユーザおよびグループ検索がNFSクライアントおよびサーバから適切なユーザおよびグループメンバーシップを返していることを確認します。

ユーザ認証

NFSマウントにアクセスすると、クライアントとサーバの間で認証方式がネゴシエートされます。標準的なAUTH_SYSアクセスの場合、ユーザクレデンシャルはプレーンテキストでネットワークを介して渡されます。つまり、ネットワーク上のスパイはこれらのクレデンシャルを参照して、それらのクレデンシャルを使用してファイルシステムにアクセスできます。標準AUTH_SYSでは、意図しないアクセスから保護するためのユーザ名/パスワードゲートウェイはありません。

NFSv3の認証で最高のセキュリティを実現するには、NFSマウントでNFS Kerberosを有効にすることを検討してください。AUTH_GSS（またはKerberos）設定では、次のようないくつかのレベルのアクセス保護が提供されます。

- Kerberos Key Distribution Centerサーバ（Windows Active Directory、FreeIPAなど）とのユーザ名/パスワードのやり取り
- 期限切れのKerberosクレデンシャルを再利用するためにクライアントの時間を変更しないように保護するセキュリティ設定。
- ログイン（krb5）、データ整合性チェック（krb5i）、エンドツーエンドのNFS暗号化（krb5p）など、NFS処理の暗号化。
- ユーザグループメンバーシップの最大数がAUTH_SYSの16からAUTH_GSSの32に増加しました。（ONTAPでは、「Auxiliary GID− Addressing the 16 GID Limitation for NFS」セクションで、この数を1、024に増やすことができます）。

詳細については、TR-4616：『NFS Kerberos in ONTAP』を参照してください。

暗号化

ONTAPは、krb5pを使用したNFSの転送中暗号化を提供しますが、保存データを暗号化するためのオプションもいくつか用意されています。次のオプションがあります。

- SEDドライブとNSEドライブ
- NVE
- NAE

暗号化とセキュリティ強化の詳細については、TR-4569：『Security Hardening Guidefor ONTAP』を参照してください。

showmount

これまで、`showmount` NFSクライアントでは、NFSサーバ上のエクスポートされたファイルシステムをユーザが参照する方法が使用されてきました。ONTAPのNFSサーバでは、デフォルトで `showmount` エクスポートされたパスを表示する機能が有効になりますが、許可されたクライアントアクセスは表示されません。代わりに、`showmount`（Everyone）にアクセス権があることが表示されます。

showmount デフォルトでは、ルートパスはコマンドに表示されません。の動作を制御し showmount、ルートパスを表示するには、次のNFSサーバオプションを使用します。

```
-showmount
-showmount-rootonly
```

この機能を使用すると、原因セキュリティスキャナが脆弱性のあるNFSサーバにフラグを付けることがあります。これは、多くの場合 showmount、返されるデータを確認するためにセキュリティスキャナがを使用するためです。そのような場合は showmount、NFSサーバでを無効にすることが必要になることがあります。

ただし、一部のアプリケーションでは、showmount Oracle OVMなどの機能にを使用します。これらのシナリオでは、セキュリティチームにアプリケーション要件を通知します。

NFSv4.xの保護

NFSv4.xはNFSv3よりも強力なセキュリティを提供するため、セキュリティを優先するすべてのNFS環境で使用する必要があります。

NFSv4.xのセキュリティ機能には次のものがあります。

- すべてのNFS処理に単一ポートで対応
- IDドメイン文字列の一致
- NFSv4.x ACL
- 統合されたKerberosサポートにより、ユーザ名とパスワードの認証とエンドツーエンドの暗号化を実現
- クライアントのスプーフィングを防止するためのクライアントIDチェック

エクスポート ポリシー ルールに関する考慮事項

NFSv4.xのエクスポートポリシールールに関する考慮事項は、一般に「NFSv3の保護」で説明している考慮事項と同じです。 これらの考慮事項を参考にしてください。

クライアントファイアウォール

NFSクライアントでは、デフォルトでファイアウォールが有効になっています（RHELではSELinux、UbuntuではAppArmorなど）。場合によっては、これらのファイアウォールがNFSの処理に影響することがあります。NFSアクセスの問題が発生した場合は、必ずクライアントのファイアウォールルールを確認し、トラブルシューティング中に必要に応じてファイアウォールサービスを停止します。

ファイアウォールポートルール

NFSv3とは異なり、NFSv4.xではすべてのNFS処理が、同じNFSポート2049を使用する複合呼び出しに統合されます。このポートはONTAPでは変更できないため、NFSv4.xを使用する予定の場合は、ポート2049を許可するようにファイアウォールポートルールを変更する必要があります。

権限

NFSv4.xでは、モードビットおよびNFSv4.x ACLを使用して権限を提供します。ACLの利点については「NFSv4 ACLを有効にするメリット」で説明しますが、NFSv4.xで最も堅牢できめ細かなセキュリティを実現するにはACLを使用します。

ユーザ認証

NFSv4.xでのユーザ認証については、「NFSv3の保護」で説明しているのと同じガイダンスに従う必要があります。最高レベルのセキュリティが必要な場合は常にKerberosを使用してください。

暗号化

NFSv4.xでの転送中および保存中の暗号化については、「NFSv3の保護」で説明しているのと同じガイダンスに従う必要があります。

ネームサービスのベストプラクティス

ONTAPは、DNS、LDAP / NIS、Kerberos、およびActive Directory用の外部ネームサービスサーバをサポートしています。NAS環境では、ネームサービスを使用すると、パフォーマンス、耐障害性、管理性に関してさまざまなメリットが得られます。ネームサービスの詳細なベストプラクティスについては、TR-4668：『Name Service Best Practices』を参照してください。ネームサービスに関する一般的なベストプラクティスは次のとおりです。

- **可能な場合は常にネームサービスを一元化します。**数百、数千のユーザ、グループ、ホスト名などが存在する環境では、組織全体でファイルのローカルコピーを同期しようとすると、不整合、ホスト名またはユーザクレデンシャルの重複、およびプラットフォーム間での一般的な混乱が発生する可能性があります。クライアントとONTAPクラスタに同じネームサービスセットを指定すると、このような環境での管理オーバーヘッドが大幅に軽減され、アクセス権限とトラブルシューティングが簡易化されます。
- **DNSは常に設定してください。**ネームサービスの一元化と同様に、ホスト名、IPアドレス、DNSゾーン、ドメイン、サービスレコードを手動で管理するのではなく、一元的に管理するサービスが不可欠です。特にDNSは、多くのNASサービス（特にWindows Active Directory、Kerberos KDC、LDAP（サービスレコードの検索とロードバランシング用）、NFSv4.x）の機能にとって重要です。
- **NFSv4.x環境でUNIX ID管理にLDAPを使用する。**NFSv4.xではユーザ認証に名前文字列が利用されるため、またサーバとクライアント間でのユーザ名の一致によってNFSv4.xの機能が制御されるため、NFSv4.xクライアントとONTAPでUNIX IDに対して一元化されたネームサービスを使用することでNFSv4.xの使用がはるかに簡単になります。
- **ネームサービスサーバを使用する場合は、複数のを指定します。**ネームサービスサーバは、毎日数千、数十万、場合によっては数百万の要求を受信する可能性があります。これらの要求には、ネットワークとネームサービスサーバの両方でパフォーマンスコストがかかります。ONTAPでは要求のキャッシュは適切に機能しますが、複数のサーバを設定してクライアントとONTAPの設定に追加しないかぎり、ネームサービスサーバに負荷がかかることがあります。さらに、同じDNS名の背後に複数のサーバをロードバランシングすることで、設定が簡単になり、ネームサービスアドレスが変更された場合にクライアントとサーバを再設定する必要がなくなり、要求のロードバランシングに使用されているサーバの数がマスクされます。

ONTAPのNASネットワークに関する一般的な考慮事項

ONTAPのNAS接続は、次の一般的なガイドラインに従っています。

- クライアントがONTAP内のデータLIFに接続すると、処理を追跡するために、そのTCP接続に一意のIDが割り当てられます。
- 同じNASクライアントが以降同じデータLIFにNAS接続要求を行うと、別のデータボリュームへのアクセスがあっても、その一意のIDが再利用されます。
- 同じNASクライアントが（同じノード上にあっても）別のデータLIFに後続のNAS接続要求を行うと、そのクライアントには新しい一意のIDが割り当てられます。
- クラスタ内の各ノードには、NASの処理に使用できるTCP接続の制限があります。この制限に達すると、リソースが解放されるまで新しい接続は拒否されます。この場合、ONTAP Event Management System（EMS；イベント管理システム）のログにエラーが表示されます。
- クラスタ内で1つのデータLIFを使用すると、着信NAS接続用のアクセスポイントは1つだけになり、シングルノードのTCP接続の制限に従います。そのため、1、000台のクライアントを1つのノードにマウントすると、4つの異なるクラスタノードに1000台のクライアントをマウントするよりも短時間でシステムリソースが枯渇するリスクがあります。EDAなどの大規模な環境では、リソースの枯渇を軽減するために、すべてのNASクライアントに単一のデータLIFを割り当てることを避けてください。
- HAペアで、各ノードに許容される最大接続IDの半分を超える数の接続IDが設定されている場合にストレージフェイルオーバーが発生すると、クライアントがNFSマウントへの接続を再確立しようとしたときに接続が失敗するため、稼働しているノードで接続IDの上限に達することがあります。フェイルオーバーが発生した場合に備えて、可能であれば、NAS接続を許容される合計制限の約50%に維持します。

- NFSv3マウントには、マウントやポートマッパーなど、接続プロセスで使用される補助プロトコルがあります。これらのUDPベースのプロトコルには、初期接続プロセス中に一意のIDが割り当てられ、60秒後に期限切れになります。マウントストームのシナリオ（何千ものクライアントが同時にマウントされている）では、補助プロトコルの一時的なID割り当てが原因で、接続リソースが人為的または時期尚早に消費される可能性があります。クライアントマウントをずらして、自動マウントのワークロードをクラスタ内の複数のノードに分散することを検討してください。また、Cloud Volumes ONTAPを使用している場合は、FlexCacheボリュームを使用する複数のCloud Volumes ONTAPインスタンスに分散することも検討してください。
- アンマウントでは、mount/portmapプロトコルの接続IDも生成されます。
- UDP接続とTCP接続の両方が、ノードの合計ネットワーク接続数にカウントされます。
- `tcp` マウントオプションとしてを指定すると、これらの余分なUDP呼び出しが排除され、各マウント/アンマウントで生成される接続の合計数が削減されます。マウント/アンマウントが大量に生成される環境では、を `tcp` マウントオプションとして使用します。
- NFSv4.xでは常にTCPが使用され、マウント/アンマウントにマウント/ポートマッパーは使用されないため、そのNFSバージョンに対して追加の接続IDは生成されません。
- 最大接続制限を超えると、EMSメッセージが生成されます（maxCID.threshold.exceededまたはmaxCID.limit.exceeded）。
- 最新のNASクライアントには、SMBマルチチャネルやNFS nConnectなど、単一のNAS接続に並列ネットワーク接続を追加する機能があります。NAS共有ごとにより多くのTCPスレッドを提供する機能を使用すると、通常はパフォーマンスが向上しますが、ONTAPのNASスタックでより多くの一意のIDが使用されます。たとえば、NFSv3の通常のマウントで使用する一意のIDは1つだけですが、nConnect=4を使用するNFSv3のマウントでは、マウントごとに最大4つの一意のIDが使用される可能性があります。EDA環境で拡張性を考慮した設計を行う場合は、この点に留意してください。
- ONTAPの各一意のIDには、同時に実行できるNAS処理の数が128に制限されています。クライアントがONTAPで管理できない数以上の同時処理を送信することでこの制限を超えた場合、リソースが解放されるまで、ONTAPのNASスタックでフロー制御が実行されます。この動作を軽減するには、クライアント側の設定を使用するか、クライアントごとに一意のIDを使用します。NASとのネットワーク同時実行の詳細については、「Network connection concurrency and TCP slots：NFSv3」を参照してください。
- ノードごとの接続IDの制限と、接続ごとのNASの同時処理数の制限に加えて、ある時点で実行可能なNAS処理の総数には、ノードレベルの制限もあります。NAS操作が実行されるたびに、その操作が完了するまでリソースコンテキストが予約されます。その時点で、リソースはシステムに解放されます。1つのノードで一度に要求されるリソースが多すぎると、パフォーマンスの問題が発生する可能性があります。これらのリソースの詳細および問題を回避する方法については、「Exec context throttling」を参照してください。

マウント/アンマウント接続動作の例

次の例は、`tcp` マウントオプションを指定した場合と指定しなかった場合のNFSv3のマウント/アンマウントで接続IDが生成される方法を示しています。これらの接続IDは、コマンドを使用して確認できます。
`network connections active show -remote-ip x.x.x.x.`

Note: `To reduce the total number of connections for NFSv3 mounts and reduce the chances of exceeding the node's connection limits, specify the tcp mount option.`

「tcp」マウントオプションを使用しないマウント/アンマウントで生成された接続ID

マウント前：

```
cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

`tcp` オプションを指定しないクライアントマウント：

```
# mount -o vers=3 DEMO:/home /mnt/client1
```

マウント後：

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                      Protocol/Service
----------- ---------------------- ---------------------------------- ----------------
Node: cluster-02
DEMO        data2:111              centos83-perf.ntap.local:56131
                                                                  UDP/port-map
DEMO        data2:635              centos83-perf.ntap.local:44961
                                                                  UDP/mount
DEMO        data2:635              centos83-perf.ntap.local:1022
                                                                  UDP/mount
DEMO        data2:2049             centos83-perf.ntap.local:879 TCP/nfs
4 entries were displayed.
```

約60秒後にUDP接続IDが消え、NFS接続だけが残ります。

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                      Protocol/Service
----------- ---------------------- ---------------------------------- ----------------
Node:  cluster-02
DEMO        data2:2049             centos83-perf.ntap.local:879 TCP/nfs
```

クライアントのアンマウント：

```
# umount /mnt/client1
```

NFS接続IDがなくなっていますが、マウント/ポートマップUDP接続が60秒以内確立されます。

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                      Protocol/Service
----------- ---------------------- ---------------------------------- ----------------
Node: cluster-02
DEMO        data2:111              centos83-perf.ntap.local:36775
                                                                  UDP/port-map
DEMO        data2:635              centos83-perf.ntap.local:33867
                                                                  UDP/mount
DEMO        data2:635              centos83-perf.ntap.local:966 UDP/mount
3 entries were displayed.

cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

「tcp」マウントオプションを使用したマウント/アンマウントで生成された接続ID

マウント前：

```
cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

tcp オプションを使用したクライアントマウント：

```
# mount -o vers=3,tcp DEMO:/home /mnt/client1
```

マウント後：

```
cluster::> network connections active show -remote-ip x.x.x.x
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                      Protocol/Service
----------- ---------------------- ---------------------------------- ----------------
Node: cluster-02
DEMO        data2:2049             centos83-perf.ntap.local:931 TCP/nfs
```

クライアントのアンマウント：

```
# umount /mnt/client1
```

NFS接続IDがなくなり、UDPマウント/ポートマッパーがありません：

```
cluster::> network connections active show -remote-ip x.x.x.x
There are no entries matching your query.
```

コンテナでのNFS接続の動作

EDA環境ではコンテナ化環境（DockerやRedHat OpenShiftなど）が普及しつつあります。そのため、NFSマウントを使用してコンテナを実装する際に、適切なサイジングとスケールアウトに関する考慮事項を確保するために、コンテナがONTAPのNFS接続にどのように影響するかを理解することが重要です。

コンテナホストにNFSマウントがある場合、各NFSマウントにはデータLIFごとに独自の接続IDが割り当てられます。NFSマウントを使用するそのホスト上のコンテナは、ホストと同じネットワークIPアドレスを共有することがよくありますが、NFSマウントを開始すると、独自の接続IDを取得します。

たとえば、次のコンテナホストにはボリュームへのマウントがあります。

```
[root@centos7-docker ~]# mount | grep scripts
demo:/scripts/dockerfiles on /dockerfiles type nfs
(rw,relatime,vers=3,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec
=sys,mountaddr=x.x.x.x,mountvers=3,mountport=635,mountproto=udp,local_lock=none,addr=x.x.x.x)

cluster::> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node                cid         vserver local-address remote-ip     proto service

clutster-01         1011516163 DEMO    x.x.x.x       x.x.x.y       TCP   nfs

cluster::> nfs connected-clients show -node * -client-ip x.x.x.y -data-lif-ip x.x.x.x

     Node: cluster-01
  Vserver: DEMO
  Data-Ip: x.x.x.x
Client-Ip       Volume-Name      Protocol Idle-Time     Local-Reqs Remote-Reqs
--------------- ---------------- -------- ------------- ---------- ------------
x.x.x.y         scripts          nfs3     55s           73         0
```

同じデータLIFへのコンテナマウントが開始されると、NFS用の新しい接続IDが、mountおよびportmap用の新しいUDP接続IDとともに生成されます。

```
[root@centos7-docker ~]# docker exec -it centos bash
[root@f8cac0b471dc /]# mount -o vers=3 10.193.67.237:/scripts /mnt

cluster::*> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node                cid         vserver local-address remote-ip     proto service
--------------------------------------------------------------------------------------------
cluster-01          1011516163 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01          1011516166 DEMO    x.x.x.x       x.x.x.y UDP   port-map
cluster-01          1011516167 DEMO    x.x.x.x       x.x.x.y UDP   mount
cluster-01          1011516168 DEMO    x.x.x.x       x.x.x.y UDP   mount
cluster-01          1011516170 DEMO    x.x.x.x       x.x.x.y TCP   nfs
```

新しいコンテナを開始してコンテナ内にマウントすると、同じデータLIF、ボリューム、ノードを使用している場合でも、同じクライアントIPから追加のNFS接続IDとUDP接続IDが作成されます。

```
[root@centos7-docker ~]# docker run --name centos2 --rm -it --cap-add SYS_ADMIN -d
parisi/centos7-secure
16dec486692dbc1133b4c1f74c6e78aa7aab875c0aed71d0d087461a6bed8060
[root@centos7-docker ~]# docker exec -it centos2 bash
[root@16dec486692d /]# mount -o vers=3 10.193.67.237:/scripts /mnt
```

```
cluster::*> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node                 cid         vserver local-address remote-ip      proto service
-------------------- ----------- ------- ------------- -------------- ----- -------
cluster-01           1011516163  DEMO    x.x.x.x       x.x.x.y TCP    nfs
cluster-01           1011516170  DEMO    x.x.x.x       x.x.x.y TCP    nfs
cluster-01           1011516177  DEMO    x.x.x.x       x.x.x.y UDP    port-map
cluster-01           1011516178  DEMO    x.x.x.x       x.x.x.y UDP    mount
cluster-01           1011516179  DEMO    x.x.x.x       x.x.x.y UDP    mount
cluster-01           1011516183  DEMO    x.x.x.x       x.x.x.y TCP    nfs
```

そのため、短時間で大量のNFSマウントを生成する環境（特にコンテナが関係している場合）では、接続IDがすぐに追加され始める可能性があります。

コンテナ環境の拡張がNFSサーバにどのような影響を及ぼすかを検討するには、次のシナリオを検討してください。コンテナホストで1、000個のコンテナが開始され、各コンテナがクラスタ内の同じデータLIFにNFSマウント要求を行います。 その1つのデータLIFで生成される接続IDの総数は、1000（`tcp` マウントオプションを指定した場合）~4、000です。 （`tcp` マウントオプションを指定しない場合、60秒以内に期限切れになる、マウント/ポートマップ用の3、000個のUDP接続ID）。

コンテナが同じノードの別々のデータLIFにマウントされている場合は、同じ接続IDの分散が適用されます。

コンテナが異なるノードの異なるデータLIFにマウントされると、接続IDが複数のノードに分散されるため、接続IDの制限に達するまでに時間がかかります。使用するクラスタノードとデータLIFの数が多いほど、分散される接続IDも多くなります。2つのノードの場合、1、000個の接続IDを1ノードあたり500に分割できます。4ノードクラスタの場合、1、000個の接続IDをノードあたり250に分割できます。

nconnectが総接続に与える影響

NFSマウントが確立されると、単一のNFS接続IDが使用されます。ただし、新しいNFSオプションを使用すると `nconnect`、 NFSマウントごとに複数のTCP接続を開くことができます。たとえば、 `nconnect=8` をNFSマウントオプションとともに使用すると、SVM内のデータLIFへのNFS接続IDが最大8つ作成されます。これによりパフォーマンスが向上しますが、予想よりも多くの接続IDが使用される可能性もあります。

例：

```
# mount -o nconnect=8 DEMO:/home /mnt/client1

cluster::*> network connections active show -remote-ip x.x.x.x -fields cid,proto,service,remote-
ip,local-address,node
node                 cid         vserver local-address remote-ip      proto service
-------------------- ----------- ------- ------------- -------------- ----- -------
cluster-02           2328843073  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843076  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843077  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843078  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843079  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843080  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843081  DEMO    x.x.x.y       x.x.x.x TCP    nfs
cluster-02           2328843082  DEMO    x.x.x.y       x.x.x.x TCP    nfs
```

マウントが大量に発生する環境でnconnectを使用する場合は、プラットフォームのノードあたりの接続IDの制限に注意してください。また、接続を適切に分散するために、複数のクラスタノードに接続を分散することを計画してください。単一のノード/データLIFをNFSマウントに使用している場合は、複数のノードを使用する場合に比べて、接続IDが大幅に不足します。接続IDの制限は、ソリューションを設計する際のスケールディスカッションに考慮する必要があります。

接続総数に対するNetApp XCPの影響

データ移行にNetApp XCPを使用する場合は、接続IDを確立するときにnconnectと同様に動作します。マウントポイントを1つ指定することもできますが、この `-parallel` オプションによってONTAPノードに対して確立される接続IDの総数が決まります。この `-parallel` オプションが定義されていない場合、XCPはデフォルトで8つのパラレル接続になります。

メモ： XCPスキャンでは、1つの接続IDのみが使用されます。

次の例では、XCPホストを使用してNFSマウントからデータをコピーします。構文は次のとおりです。

```
# xcp copy -parallel 16 IP1,IP2:/files destination:/files
```

上記のコマンドでは、XCPは指定されたIPアドレスごとに16のネットワークスレッドを作成しています。この場合、IP1とIP2はソースクラスタの同じノードにあります。その結果、このノードは、このジョブに対して確立された合計34の接続IDを取得します（指定した最初のIPでは18、2番目のIPでは16）。

```
cluster::*> network connections active show -remote-ip x.x.x.y -fields cid,proto,service,remote-
ip,local-address,node
node                 cid         vserver local-address remote-ip     proto service
-------------------- ----------- ------- ------------- ------------- ----- -------
cluster-01           1011516323 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516327 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516328 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516329 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516330 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516331 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516332 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516333 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516334 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516335 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516336 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516337 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516338 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516339 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516340 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516342 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516343 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516344 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516345 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516346 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516347 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516348 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516349 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516350 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516351 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516352 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516353 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516354 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516355 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516356 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516357 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516358 DEMO    x.x.x.z       x.x.x.y TCP   nfs
cluster-01           1011516359 DEMO    x.x.x.x       x.x.x.y TCP   nfs
cluster-01           1011516360 DEMO    x.x.x.z       x.x.x.y TCP   nfs
34 entries were displayed.
```

XCPを使用する場合は、複数のノード上の複数のIPアドレスに接続を分散してみてください。並列スレッドの数によって、使用されているネットワーク接続IDの総数が増えることに注意してください。

接続IDの最大値と割り当ての表示

ONTAPでは、ノードで使用可能な接続ID（CID）の数、現在使用中の接続IDの数、それらの接続を使用しているクライアントを確認できます。

接続ID情報を表示するには、次のコマンドに注意してください。

ネットワーク接続がアクティブ（admin権限）

```
cluster::> network connections active ?
  delete                      *Delete an active connection in this cluster
  show                        Show the active connections in this cluster
  show-clients                Show a count of the active connections by client
  show-lifs                   Show a count of the active connections by logical interface
  show-protocols              Show a count of the active connections by protocol
  show-services               Show a count of the active connections by service
```

statistics start / show -object cid（diag権限）

```
cluster::*> statistics start -object cid
```

node run netstat -na（diag権限）

```
cluster::*> node run -node node1 netstat -na
```

統計を使用した接続IDの動作と最大値の例

このオブジェクトの統計を収集するには、diagnostic権限で次のコマンドを実行します。

```
cluster::*> statistics start -object cid
```

sample-id：sample_55の統計収集を開始しています

出力を表示するには：

```
cluster::*> statistics show -object cid
```

次に、NFSマウントが確立される前にこれらの結果が表示される例を示します。

```
Object: cid
Instance: cid
Start-time: 6/25/2021 16:54:27
End-time: 6/25/2021 16:56:37
Elapsed-time: 130s
Scope: cluster-01

    Counter                                                     Value
    -------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                     0
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        352
    in_use_max                                                      0
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526
```

この出力では、ノードのは `cid_max` 115904です。このうち、10526は `reserved_cid` ONTAPシステム処理用であり、クライアント処理に使用できる接続IDの合計数は115904-10526 = 105378です。ノードがこの制限を超えると、EMSはmaxCID.limit.exceededメッセージをトリガーします。

現在、352個の `in_use` CIDが存在する。

`nconnect=8` ノード1でを使用したNFSマウントが確立されると、次のように数値が変わります。

```
Object: cid
Instance: cid
Start-time: 6/25/2021 16:54:27
```

```
End-time: 6/25/2021 17:04:54
Elapsed-time: 627s
Scope: cluster-01

    Counter                                                     Value
    -------------------------------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                    14
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        360
    in_use_max                                                      0
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526
```

現在、_useで352個のcidが使用されている代わりに、360個のcidが使用されています。これにより、nconnectマウントで作成された8つのアクティブなネットワーク接続が調整されます。

tcp マウントオプションを指定せずにNFSv3マウントを実行すると、4つの新しいCIDが作成され in_use 、統計で確認できます。

```
    Counter                                                     Value
    -------------------------------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
    alloc_total                                                    28
    cid_max                                                    115904
    execs_blocked_on_cid                                            0
    in_use                                                        364
    in_use_max                                                      1
    instance_name                                                 cid
    node_name                                                 cluster-01
    process_name                                                    -
    reserved_cid                                                10526
```

これらのCIDには network connections active show 、コマンドで表示されるmountおよびportmap UDPエントリが含まれます。

```
cluster::*> network connections active show -remote-ip x.x.x.x -fields cid,proto,service,remote-
ip,local-address,node
node                 cid        vserver local-address remote-ip     proto service
-------------------- ---------- ------- ------------- ------------- ----- -------
cluster-01           1011516253 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516256 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516257 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516258 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516259 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516260 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516261 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516262 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516268 DEMO    x.x.x.z       x.x.x.x UDP   port-map
cluster-01           1011516269 DEMO    x.x.x.z       x.x.x.x UDP   mount
cluster-01           1011516270 DEMO    x.x.x.z       x.x.x.x UDP   mount
cluster-01           1011516272 DEMO    x.x.x.z       x.x.x.x TCP   nfs
```

UDPエントリが期限切れになると（約60秒後）、新しい接続で使用できるようにCIDが解放され、データLIF IP **x.x.x.zのNFS接続で使用される余分なCIDは1つだけになり**ます。

```
    Counter                                                     Value
    -------------------------------------------------------- --------------------------------
    alloc_failures_nomem                                            0
    alloc_failures_reserved_toomany                                 0
    alloc_failures_toomany                                          0
```

```
    alloc_total                                                  28
    cid_max                                                  115904
    execs_blocked_on_cid                                          0
    in_use                                                      361
    in_use_max                                                    1
    instance_name                                               cid
    node_name                                           cluster-01
    process_name                                                  -
    reserved_cid                                              10526

cluster::*> network connections active show -remote-ip x.x.x.x -fields cid,proto,service,remote-
ip,local-address,node
node                 cid         vserver local-address remote-ip     proto service
-------------------- ----------- ------- ------------- ------------- ----- -------
cluster-01           1011516253 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516256 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516257 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516258 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516259 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516260 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516261 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516262 DEMO    x.x.x.y       x.x.x.x TCP   nfs
cluster-01           1011516272 DEMO    x.x.x.z       x.x.x.x TCP   nfs
```

「rootonly」オプション-nfsrootonlyおよびmountrootonly

rootonly 信頼されていないクライアントアクセスを防止するためのオプションが追加されました。信頼されていないクライアント（エクスポートルールに含まれていないクライアント）は、信頼されたクライアントへのSSHトンネリングを使用してデータにアクセスする可能性があります。ただし、これらの要求は信頼できないポート（1、024より大きいポート）から送信されます。これは、アクセスする意図のないクライアントにとってバック ドアとなります。

そのため、 rootonly オプションの有効化と無効化は必要に応じて異なります。つまり、環境でNFSが正常に機能するためにより多くのポートが必要か、それとも信頼できないクライアントによるマウントへのアクセスを防止する方が重要かということです。

NFSv4.xやKerberos認証を利用してNFSエクスポートへのアクセスのセキュリティを強化することも1つの妥協案です。TR-4616：『NFS Kerberos in ONTAP』では、NFS Kerberosの使用方法について説明しています。

このようなシナリオでは mount-rootonly、nfs-rootonly オプションやオプションを使用することで問題を軽減できます。

クライアントのポート使用状況を確認するには

```
# netstat -na | grep [IP address]
```

クラスタでのポートの使用状況を確認する方法

```
cluster::> network connections active show -node [nodename] -vserver [vservername] -service nfs*
```

たとえば、マウントオプションを使用して、クライアントが予約されたポート範囲外のポートを使用するように指定でき noresvport ます（resvport 指定しない場合はデフォルトで、1～1024のソースポートを使用します）。この処理を実行し、 mount-rootonly NFS SVMに対してを有効にすると、マウントが失敗します。

```
cluster::*> nfs show -vserver DEMO -fields mount-rootonly,nfs-rootonly
vserver mount-rootonly nfs-rootonly
------- ---------------- ----------------
DEMO    enabled          disabled

# mount -o noresvport,vers=3 demo:/scripts /mnt/client1
mount.nfs: access denied by server while mounting demo:/scripts
```

パケットトレースから、クライアントの送信元ポートが許容範囲（36643）外にあることがわかります。

```
129    x.x.x.x        x.x.x.y        MOUNT   138    V3 MNT Call (Reply In 130) /scripts
User Datagram Protocol, Src Port: 36643, Dst Port: 635
```

を使用する resvportと、マウントは成功します。

```
# mount -o resvport,vers=3 demo:/scripts /mnt/client1
#
```

パケットトレースは、送信元ポートが1024の範囲（703）内にあることを示しています。

```
44      x.x.x.x                         MOUNT   138     V3 MNT Call (Reply In 45) /scripts
User Datagram Protocol, Src Port: 703, Dst Port: 635
45      x.x.x.y         x.x.x.x         MOUNT   150     V3 MNT Reply (Call In 44)
```

この mount-rootonly オプションをdisabledに設定すると、次のようになります。

```
cluster::*> nfs modify -vserver DEMO -mount-rootonly disabled
```

を使用したマウントは noresvport 成功します。

```
# mount -o noresvport,vers=3 demo:/scripts /mnt/client1
#
```

トレースは、送信元ポートが1024の範囲（58323）から外れていることを示しています。

```
101     x.x.x.x         x.x.x.y         MOUNT   138     V3 MNT Call (Reply In 102) /scripts
User Datagram Protocol, Src Port: 58323, Dst Port: 635
102     x.x.x.y         x.x.x.x         MOUNT   150     V3 MNT Reply (Call In 101)
```

注： NFSv4.xマウントではNFSマウントに補助マウントプロトコルは使用されないため、 mount- rootonly ポートはこれらの処理には影響せず、NFSv3マウントにのみ影響します。

多数のNFSクライアントでマウントポートが枯渇している

多数のクライアントがNFS経由で接続している環境では、マウントポートの数がデフォルトで1、024に制限されていることに注意してください。「マウントストーム」のシナリオ（何千ものクライアントがほぼ同時にマウントまたはアンマウントを実行）では、1、024個のポートがすぐに使い果たされ、接続の問題が発生する可能性があります。

- マウント処理（NFSv3のみ）は、NFSオプション（ mount-rootonlyデフォルトでenabledに設定）で制限されます。これにより、マウントの着信ポート範囲が1～1024に制限されます。
- ポート2049を介したNFS処理の場合、許可される受信ポートのデフォルト数は65、534です。これは nfs-rootonly nfsオプションで制御され、 デフォルトでdisabledに設定されています。受信NFSクライアントのソースポートの数を1、024に制限するには、このオプションを **enabledに設定**します。

場合によっては、マウントまたはNFS処理に使用されるポートの数が不足し、ポートが使用可能になるまで後続のマウントおよびNFS処理がハングしたり失敗したりすることがあります。

環境内に数千ものクライアントがNFS経由でマウントされ、I/Oを生成している場合（「コンテナとのNFS接続の動作」のコンテナの例など）、NFSサーバのすべてのポートが使用されている可能性があります。たとえば、ESXでNFSデータストアを使用するケースがあります。これは、一部の従来のベストプラクティスではデータストアごとにデータLIF / IPアドレスが必要であるためです。そのため、多数のボリュームやデータストアがある環境では、NFSポートがオーバーラン状態になります。この場合は mount-rootonly nfs-rootonly 、NFSサーバでオプションやオプション（あるいはその両方）を無効にして対処します。
この処理を実行すると、1~1、024のポート範囲の制限が解除され、NFSサーバでマウントやNFS処理に使用できるポートが最大65、534個になります。noresvport NFSマウントは resvport 指定しない場合はデフォルトでマウントオプションが使用されるため、クライアント側ではマウントオプションを使用して非権限のソースポートを使用することも必要になる場合があります。

UDPとTCPでのマウントポートの用途

デフォルトでは、NFSv3のマウントではMOUNTプロトコルにUDPが使用されます。UDPには確認応答がないため、ONTAPはUDPマウントの接続IDを60秒間保持し、ONTAPが接続IDを削除する前にマウントが成功または失敗する可能性を確保します。環境のアンマウント処理も実行します。

このため、大量のマウント/アンマウント要求が同時に発生すると、60秒の有効期限に達して既存のマウント接続IDがクリアされるまで、1～1024の範囲で使用可能なすべてのソースポートが使用される可能性があります。UDPを使用したマウントに失敗した場合でも、接続IDは60秒間維持されます。

たとえば、許可された範囲外のポートを使用したマウントは失敗しました。

```
# mount -o noresvport,vers=3 demo:/scripts /mnt/client1
mount.nfs: access denied by server while mounting demo:/scripts
```

ONTAPでは、失敗したマウント要求によって生成された3つの接続IDが60秒後に期限切れになることを確認できます。マウントストームでは、60秒は永遠に続くことがあります。

```
cluster::*> network connections active show -remote-ip x.x.x.x -fields remote-port,cid,service
node                 cid         vserver remote-port service
-------------------- ----------- ------- ----------- ----------
cluster-02           2328843541 DEMO    35470       port-map
cluster-02           2328843542 DEMO    60414       mount
cluster-02           2328843543 DEMO    33649       mount
3 entries were displayed.
```

TCPの通信では確認応答が使用されるため、マウント要求が成功したあともONTAPで接続IDを維持する必要はありません。そのため、を `-o tcp` NFSマウントオプションに使用すると、1~1024の範囲のポートが短時間で解放され、マウントでポートが不足するほとんどのケースを回避できます。

多数のNFSクライアントを使用する環境のベストプラクティス

次のセクションでは、EDAコンピュートファームなど、多数のNFSクライアントを含む環境のベストプラクティスをいくつか紹介します。

マウントストームが発生する環境に関する考慮事項

マウントストームとは、多数のNFSクライアントが短期間に同じNFSサーバまたはクラスタに対してNFSエクスポートをマウントまたはアンマウントすることです。自動マウントツールを頻繁に使用するNFS環境では、管理者が意識することなくマウントストームが発生する可能性があります。

マウントストームのシナリオでは、システムリソースを短時間で使い切ることができます。マウントストームが発生する可能性がある環境を設計する際には、次のベストプラクティスを考慮する必要があります。

- 各ノードのクラスタノードとデータLIFの数が多いほど、NFSマウントや接続などの負荷を分散するために使用できるシステムリソースが増えます。大規模なNFSマウント環境を設計する場合は、受信するNFS接続にできるだけ多くのノードを使用することを検討してください。ONTAPは、NASのみのクラスタで最大24ノードをサポートします。
- 1つのクラスタで複数のデータLIFを使用して接続のロードバランシングを行う場合は、DNSロードバランサを使用して1つのホスト名で多数のIPアドレスをマスクすることで環境を簡易化します。これにより、接続の変更をエンドユーザに通知することなく、DNS FQDNにIPアドレスを簡単に追加/削除でき、自動マウントが使用されているかどうかにかかわらず、クライアントからの着信NAS接続のバランスを均等に維持できます。
- データLIFの場所とストレージフェイルオーバーを綿密に監視します。データLIFを別のノードに移行する場合（ポート障害または手動移行が原因で）、そのデータLIFは現在のノードのリソースをNAS接続に使用しており、リソースを短時間で枯渇させる可能性があります。ストレージフェイルオーバー（計画的または計画外）についても同様です。障害が発生したノードのデータLIFは稼働しているパートナーに移行され、そのノードがすべてのNAS接続を維持する必要があるためです。データLIFの移行が発生した場合は、移行の原因となった問題を解決し、データLIFをホームノードにリバートします。
- `statistics start -object cid` または同等のREST API機能を使用して、クラスタ内の接続IDの使用状況を監視します。`in_use max_cid` ストレージフェイルオーバーが発生した場合に備えて、接続IDの値をノードの値の約50%（`in_use` ノードあたり約50、000 ID）にしてください。そのためには、新しいデータLIFを持つノードをクラスタに追加します。
- EMSでイベントを監視し `maxCID.limit.exceeded`、イベント

maxCID.threshold.exceedednblade.execsOverLimit, が生成される場合はNetAppサポートにお問い合わせください。

- ノードあたりのデータLIF数やnconnectを使用してNFS接続を多重化すると、一部のワークロードのパフォーマンスが向上しますが、ストレージノードあたりの使用可能なリソース（接続IDやEXECコンテキストなど）も増えます。多重化を使用する場合は、潜在的な副作用に注意してください（「総接続数に対するnconnectの影響」および「Execコンテキストスロットル」を参照）。
- 可能であれば、 tcp NFSマウントにマウントオプションを使用します（-o tcp）。これにより、マウント/アンマウント処理ごとに生成される接続IDの総数が少なくなり、マウントストームのシナリオでリソースが不足する（接続IDとNFSマウントポート）を防ぐことができます。
- 使用可能な受信NFSマウントポートが1024個を超える場合（たとえば、2000個のクライアントが同時にマウントする場合）は、NFSサーバオプションを無効にし mount- rootonly、NFS noresvport クライアント/自動マウントでマウントオプションを使用し、 tcp マウントごとに生成される接続IDの数を減らすことを検討してください。（UDPマウント接続は最大60秒間キャッシュに保持されます。TCPマウント接続はクライアントACKで削除されます）。

注：NFSv4.xはUDPおよびMOUNTプロトコルの問題の影響を受けませんが、「NFSv4.xの考慮事項」で説明するように独自の課題があります。

NFSクライアントのベストプラクティス

NFSクライアントのベストプラクティスは、一般に使用するアプリケーションによって異なります。NFSクライアントの設定方法を決定する際は、必ずアプリケーションベンダーに相談してください。以下に示すベストプラクティスの推奨事項は実際には設定されておらず、アプリケーションの推奨事項やワークロードテストによって上書きすることができます。

クライアントとNFSユーティリティのバージョン

NFSクライアントのバージョンの場合は、通常、最新のOSパッチバージョンを実行し、NFSユーティリティを最新リリースに更新して最新のバグ修正と機能を適用することを推奨します。OSユーティリティとNFSユーティリティのバージョンが認定され、標準化されている環境では、この方法を選択することはできません。

原則として、ONTAP NFSはRFC標準に準拠するすべてのNFSクライアントをサポートします。これには、カスタムカーネルコンパイルを持つNFSクライアントは含まれません。

RPCスロットテーブル

RPCスロットテーブルは、NFSクライアントとサーバで許可される単一のTCP接続で許可される最大スレッド数です。これらの値は、NFSクライアント上のsunrpc設定によって制御されます。最新のNFSクライアントバージョンのデフォルトの動的スロットテーブル値は65536です。つまり、クライアントは1つのTCP接続でできるだけ多くのスロットテーブルを使用しようとします。ただし、ONTAPでは、TCP接続ごとに128スロットテーブルしかサポートされません。クライアントがこの値を超えると、ONTAPはNASフロー制御を実行し、リソースが解放されるまでクライアントの処理を一時停止します。

ベストプラクティスとして、NFSクライアントではスロットテーブルの値を128以下の静的な値に設定することを推奨します。多数のクライアントがある環境では、この値を16に設定する必要があります。パフォーマンスへの影響の詳細などの詳細については、「Network connection concurrency and TCP slots：NFSv3」を参照してください。

マウント オプション

NFSマウントオプションの推奨事項は、使用するワークロードとアプリケーションのみによって異なります。特定のマウントオプションに関する一般的なアドバイスはありますが、どのNFSオプションを使用するかは、クライアントOS管理者とアプリケーションベンダーの推奨事項によって決まります。マウントオプションに関する推奨事項は1つだけではありません。以降のセクションでは、NFSマウントオプションの一部についてのみ説明します。サポートされているNFSのマウントオプションの一覧については man nfs、NFSクライア

ントでを使用してください。

デフォルトのマウントオプション

Linuxクライアントでは、デフォルトのマウントオプションがあらかじめ設定されています。これらのデフォルトオプションは、クライアントOSのバージョンと、クライアントで検出されたNFS構成ファイルによって異なります。デフォルトのマウントオプションは、-oフラグでオプションが指定されていないマウント操作中に設定されます。

場合によっては、マウントオプションがNFSサーバとネゴシエートされます。特に新しいLinuxカーネルでは、rsize/wsizeの値とNFSのバージョンはNFSサーバの設定に基づいています。

たとえば、NFSv4.1が有効になっていて、構成ファイルまたはmountコマンドにNFSバージョンが指定されていない場合、クライアントはNFSv4.1を使用します。これは、サポートされている最も高いNFSバージョンであるためです。

次の例は、特定のオプションを実行しなかったmountコマンドの出力を示しています。ONTAP NFSサーバでTCP最大転送サイズ（`-tcp-max-xfer-size`）を1MBに設定し、NFSv4.1が有効になっています。

```
# mount DEMO:/flexgroup_16 /flexgroup
# mount | grep flexgroup
DEMO:/flexgroup_16 on /flexgroup type nfs4
(rw,relatime,vers=4.1,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,port=0,timeo=600,retr
ans=2,sec=sys,clientaddr=10.x.x.x,local_lock=none,addr=10.x.x.y)
```

wsize/rsize

マウントオプションのwsizeとrsizeによって、送信パケットごとにNFSクライアントとサーバ間で送信されるデータの量が決まります。これは、特定のアプリケーションのパフォーマンスを最適化するのに役立ちますが、アプリケーションベンダーのベストプラクティスに従って設定する必要があります。1つのアプリケーションに最適なものは、他のアプリケーションに最適ではない場合があるためです。

新しいNFSクライアントでは `-tcp-max-xfer-size` 、mountコマンドで明示的に値が設定されていない場合、wsizeとrsizeの値がONTAP NFSサーバで設定されている値に自動ネゴシエーションされます。ONTAPのデフォルト `-tcp-max-xfer-size` は64Kで、最大1MBに設定できます。

注： の一般的な推奨事項 `-tcp-max-xfer-size` は、ONTAPの値を262144（256K）に増やし、アプリケーションで必要に応じて明示的なマウントオプションを指定することです。

ワークロードタイプやwsize / rsizeの値が異なる場合のパフォーマンステストの例については 、「TCP最大転送ウィンドウサイズが異なる場合のパフォーマンスの例」を参照してください。

特定のアプリケーションに対するwsize/rsizeの推奨事項の例については、次を参照してください。

- TR-3633：『Data ONTAPを基盤にしたOracleデータベース』
- TR-4435：『SAP HANA on NetApp AFF Systems using NFS』

NFS先読み

NFSの先読みは、NFSクライアントがファイルのブロックを予測的に要求することで、シーケンシャルI/Oワークロードのパフォーマンスとスループットを向上させる方法です。最近まで、NFSマウントの先読み値は、マウントのrsize値の15倍に設定されていました。たとえば、rsizeを64KiBに設定すると、先読みサイズは960KiBになります。

最新のNFSクライアント（RHEL 8.3以降やUbuntu 18.04以降など）では、先読み値はmount rsizeで決まらなくなりましたが、グローバルなデフォルト値は128KiBです。これにより、読み取り時に原因のパフォーマンスが大幅に低下する可能性があります。デフォルトの128KiB先読み値を使用する新しいバージョンのLinuxクライアントでは、この値をより高い制限値に設定することを推奨します。異なる値を使用して読み取りパフォーマンスをテストすることが推奨されますが、内部NetAppテストでは、シーケンシャル読み取りのワークロードでは、この値を15360KiBまで安全に設定できることがわかりました。

先読み値の設定と表示の詳細については、クライアントOSベンダーにお問い合わせください。たとえば、次のSUSE KBでは、これらのOSクライアントの先読みについて説明しています。Tuning NFS client read ahead on SLE 10 and 11。

CentOS/RedHatクライアントの場合も同様です。

```
# cat /etc/redhat-release
CentOS Linux release 7.8.2003 (Core)
```

次のコマンドを使用して、マウントのBDEV情報を検索します（BDEV形式はN：NN）。

```
# cat /proc/self/mountinfo | grep /mnt/client1
125 39 0:46 / /mnt/client1 rw,relatime shared:107 - nfs DEMO:/files
rw,vers=3,rsize=1048576,wsize=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,mount
addr=10.193.67.219,mountvers=3,mountport=635,mountproto=udp,local_lock=none,addr=10.193.67.219
```

BDEV情報を使用して先読み値（そのマウントポイントのBDEVは0：46）を確認します。

```
# cat /sys/class/bdi/0:46/read_ahead_kb
15360
```

上記の例では、`/mnt/client1` CentOS 7.8クライアントでは、先読みがマウント用に15360KiB（rsizeの15倍）に設定され、rsizeが1MBに設定されています。

CentOS 8.3では、マウントがデフォルトで設定される値です。

```
# cat /sys/class/bdi/0:50/read_ahead_kb
128
```

ActimeoとNocto

NFSv3は、キャッシュされたファイル/ディレクトリデータとキャッシュされたファイル/ディレクトリ属性を使用して、共有ファイルシステムの整合性とアプリケーションのパフォーマンスを管理します。アプリケーションが共有ストレージにアクセスしてデータを使用するたびに取得する必要がないため、整合性が緩いことになります。これは、アプリケーションのパフォーマンスに大きな影響を与える可能性があります。キャッシュされた情報には、キャッシュデータを信頼する期間を設定するタイマーがあり、タイムアウト時に、次のタイムアウトまでデータを再検証するための軽量で高速なgetattr/access呼び出しがあります。

このプロセスを管理するメカニズムは2つあります。

- **CTO：**クローズとオープンの整合性により、キャッシュに関係なくファイルの最新データを取得できます。
- **Actimeo。**属性キャッシュタイマー（ファイルのデフォルトは3秒、ディレクトリのデフォルトは30秒）。

クライアントがデータの完全な所有権を持っている場合（共有されていない場合など）は、一貫性が保証されます。ストレージに対する属性取得/アクセス操作を減らし、CTO整合性をオフにし（マウントオプションとしてnocto）、属性キャッシュ管理のタイムアウトをオンにすることでアプリケーションの速度を上げることができます（マウントオプションとしてactimeo=600を指定すると、タイマーは前述のデフォルト値に対して10mに変更されます）。いくつかのテストでは、noctoはgetattr/access呼び出しの65～70%を削減し、actimeoはさらに20～25%を削減します。

クライアントによる完全な所有権がない場合でも、同様のマウントオプションを使用することでメリットが得られるケースもあります。EDA、Webホスティング、ムービーレンダリングなどのクライアントのグリッドを使用し、比較的静的なデータセット（EDAのツール/ライブラリ、WebホスティングのWebコンテンツなど）を持つアプリケーションの場合 典型的な動作は、データセットがクライアントに大部分キャッシュされることです(読み取りはほとんどなく、書き込みはありません)。このような場合、getattr/access呼び出しがストレージに返されます。これらのデータセットは、通常、ファイルシステムをマウントして定期的にコンテンツの更新をプッシュする別のクライアントによって更新されます。場合によっては、更新のために複数のファイルシステムにSnapMirror関係をプッシュすることもできます。

このような場合、新しいコンテンツの取得には既知の遅延がありますが、アプリケーションはまだ更新されていない可能性のあるデータで動作します。このようなシナリオでは、noctoとactimeoを使用して、データ不足

の日付を管理できる期間を制御できます。たとえば、ツールやライブラリ、その他の静的コンテンツを使用するEDAでは、actimeo=600は通常このデータが更新されないため、うまく機能します。クライアントがサイトの編集中にデータ更新のタイミングを確認する必要がある小規模なWebホスティングの場合、actimeo=10が許容される可能性があります。コンテンツが複数のファイルシステムにプッシュされている大規模なWebサイトでは、actimeo=60の方が効果的な場合があります。いつものように、環境でテストしてください。

このマウントオプションを使用すると、これらのインスタンスのストレージへのワークロードが大幅に削減されます（たとえば、最近のEDAでツールボリュームのIOPSが15万以上から6、000以下に低下した場合など）。また、メモリ内のデータを信頼できるため、アプリケーションの実行時間が大幅に短縮されます。NFSストレージを照会する必要がなくなります。これは、ONTAPノードの全体的なCPUと負荷を削減するのにも役立ちます。

アクティメオ

actimeoマウントオプションは、NFSクライアントでの属性キャッシュタイムアウトを制御します。actimeoオプションは、次のような使用可能な属性キャッシュの全範囲を対象としています。

```
acregmin=n
The minimum time (in seconds) that the NFS client caches attributes of a regular file before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 3-second minimum.

acregmax=n
The maximum time (in seconds) that the NFS client caches attributes of a regular file before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 60-second maximum.

acdirmin=n
The minimum time (in seconds) that the NFS client caches attributes of a directory before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 30-second minimum.

acdirmax=n
The maximum time (in seconds) that the NFS client caches attributes of a directory before it
requests fresh attribute information from a server. If this option is not specified, the NFS
client uses a 60-second maximum.
```

属性キャッシュは、メタデータ呼び出しの数を減らすことで、ネットワークをいくらか軽減します。また、一部のワークロードではメタデータ処理をクライアントでローカルに実行できるため、レイテンシの低減にも役立ちます。属性キャッシュは、ストレージに対するすべての処理がメタデータ（特にアクセス呼び出し）である場合を除き、一般に処理全体の数には影響しません。

たとえば、お客様向けコンセプトの実証（CPOC）ラボでは、actimeoを10分（600秒）に設定し、vdbenchで生成されたEDAワークロードでレイテンシを半分に（2.08ミリ秒~1.05ミリ秒）削減しました。図13 はactimeoのデフォルトレイテンシー、図14 はactimeo 600レイテンシーを示しています。

図13）デフォルトのactimeoレイテンシー–vdbench

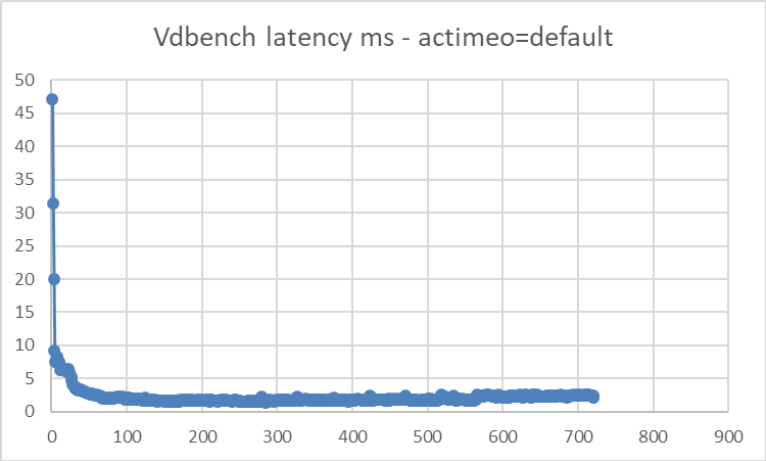
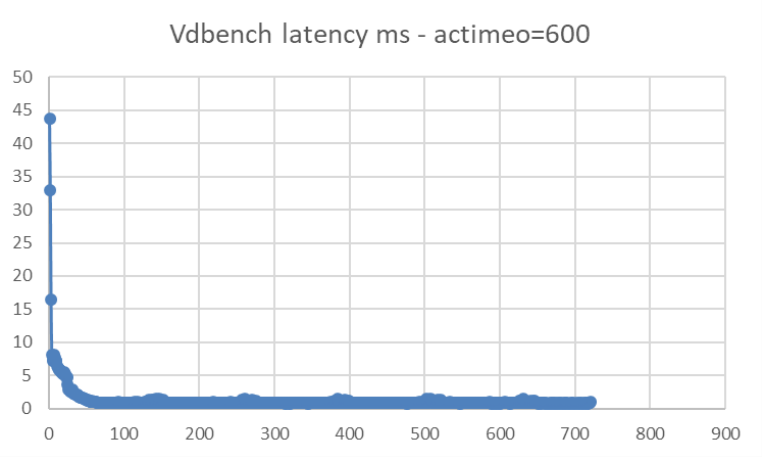


図14）Actimeo = 600のレイテンシ–vdbench



actimeo値を高く設定しすぎると、キャッシュタイムアウトが発生するまで変更された属性が適切に反映されず、予期しないアクセスの問題が発生する可能性があります。

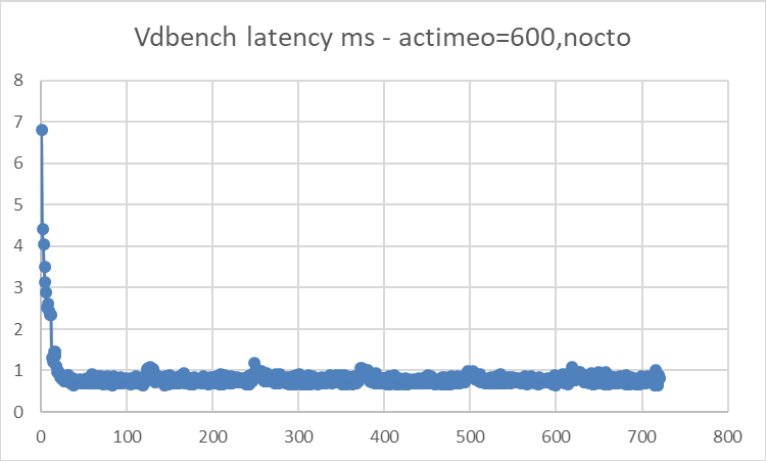
注: 属性キャッシュの推奨事項は、アプリケーションベンダーから特に指示がない限り、またはテストでパフォーマンスが大幅に向上した場合を除き、デフォルトのままにすることです。

ノクト

NOCTOは、NO CLOSE-TO-OPENの略で、書き込みが完了する前にファイルを閉じることで時間を節約できます。これがNFS環境で意味するのは、読み取り用にファイルを開いている他のクライアントでは、そのファイルに対する整合性のある更新が行われないことです。デフォルトでは、NFSマウントではnoctoオプションは設定されていません。つまり、すべてのファイルが書き込みの完了を待機してからクローズを許可します。

noctoオプションは、主にrawパフォーマンスを向上させるために使用されます。たとえば、Customer Proof of Concept Labsで実行したvdbenchテストでは、noctoマウントオプションによってレイテンシがさらに.35mSから.7msに低減されました（図15を参照）。

図15）actimeo=600、nocto latency−vdbench



注： noctoオプションの推奨事項は、読み取り負荷が高い/読み取り中心のワークロード、または複数のシステム間でデータが共有されていないワークロード（シングルライターワークロードなど）でのみ使用することです。

clientaddr

デフォルトでは、clientaddrマウントオプションは、NFSクライアントのIPアドレスを使用して自動的に設定されます。ただし、場合によっては、NFSマウントでこのオプションを指定しなければならないことがあります。

clientaddrを指定する必要がある場合は、次の2つのシナリオが考えられます。

- 必要なIPアドレスをNFSが接続に使用するようにするには、マルチNICクライアントでこのオプションの指定が必要になる場合があります。
- **NFSv4.xクライアントで** 、ホスト名が同じ（IPアドレスが異なる）2つのクライアントが同じSVM内のNFSエクスポートにアクセスしようとする場合は、このオプションの指定が必要になることがあります。NFSv4.xは、ホスト名に基づいてクライアントIDをNFSサーバに送信します。ONTAPはで応答し `CLID_IN_USE` 、同じクライアントIDを使用している場合は2番目のクライアントがマウントされないようにします。clientaddrオプションを指定すると、以降のマウント試行時にクライアントIDが増分されることがあります。

  注: ほとんどの場合、clientaddrオプションを指定する必要はありません。

nconnect

nconnectという新しいNFSマウントオプションは、NFSマウントで使用する初期段階にあります。nconnectオプションは、新しいLinuxクライアントでのみ使用できます。カーネルでオプションがサポートされているかどうかを確認するには、OSベンダーのマニュアルを参照してください。

nconnectの目的は、クライアント上のTCP接続またはマウントポイントごとに複数のトランスポート接続を提供することです。これにより、NFSマウントの並列処理能力とパフォーマンスが向上します。nconnectの詳細と、nconnectがCloud Volumes ONTAPでNFSのパフォーマンスを向上させる方法については、ブログ記事 The Real Baseline Performance Story: NetApp Cloud Volumes Service for AWSを参照してください。

ONTAP 9.8以降では、NFSマウントでnconnectを使用することが正式にサポートされています（NFSクライアントでもnconnectがサポートされている場合）。nconnectを使用するには、クライアントのバージョンがnconnectを提供しているかどうかを確認し、ONTAP 9.8以降を使用してください。ONTAP 9.8以降では、オプションなしでデフォルトでnconnectがサポートされます。

注： NFSv4.0ではnconnectの使用は推奨されません。NFSv3、NFSv4.1、NFSv4.2はnconnectで正常に動作するはずです。

表15 に、異なるnconnectスレッド値を使用した単一のUbuntuクライアントの結果を示します。

表15）nconnectのパフォーマンス結果

nconnect値	プロセスあたりのスレッド数	スループット	差異
1	128	1.45GB/秒	-
2	128	2.4GB/秒	+66%
4	128	3.9GB/秒	+169%
8	256	4.07GB/秒	+181%

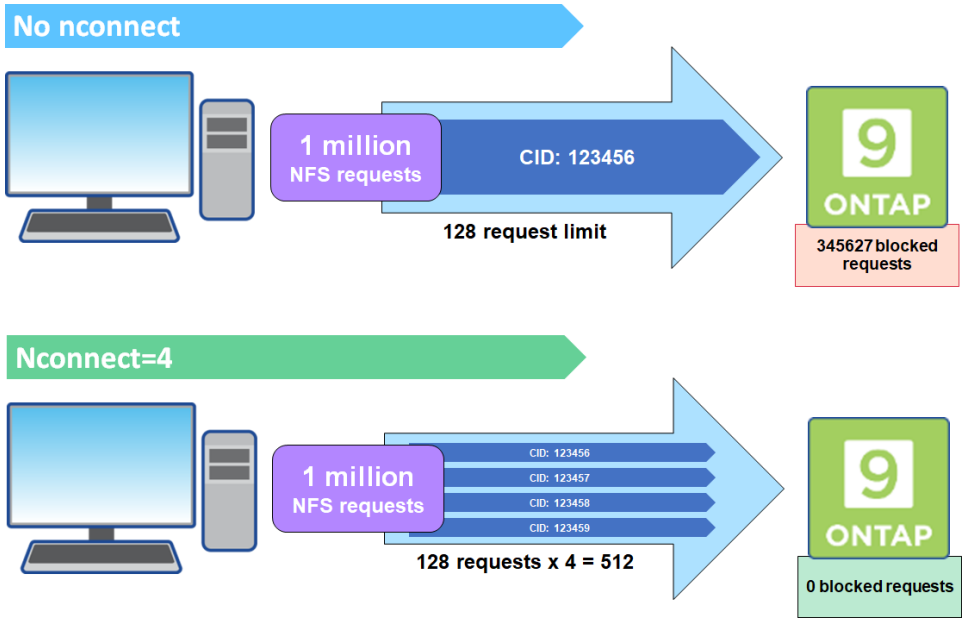
注： nconnectの使用を推奨するかどうかは、クライアントOSとアプリケーションのニーズによって異なります。本番環境に導入する前に、この新しいオプションを使用してテストすることを強く推奨します。

nconnectが動作していることを確認する方法を教えてください。

nconnectは、1つのTCP接続でより多くのセッションを割り当てるように設計されています。これにより、NFSワークロードをより適切に分散し、接続に並列処理を追加して、NFSサーバがワークロードをより効率的に処理できるようになります。ONTAPでは、NFSマウントが確立されると、Connection ID（CID；接続ID）が作成されます。このCIDは、最大128の同時実行中操作を提供します。クライアントがこの数を超えると、ONTAPはフロー制御を実行し、他の処理が完了した時点で利用可能なリソースの一部を解放します。通常、この一時停止はわずか数マイクロ秒ですが、数百万回の処理が行われると、結果としてパフォーマンスの問題が発生する可能性があります。nconnectは128の制限を取り、クライアントのnconnectセッション数に掛けることができます。これにより、表15に示すように、CIDあたりの同時処理数が増加し、パフォーマンスが向上する可能性があります。

図16 は、nconnectを使用しないマウントでの同時処理の処理と、nconnectが処理をNFSマウントに分散する仕組みを示しています。

図16）nconnectを使用したNFSマウントと使用しないNFSマウント



お使いの環境でnconnectが実際に機能しているかどうかを確認するには、いくつかの点を確認します。

nconnectを使用しない場合は、クライアントマウントごとに1つのCIDが確立されます。これらのCIDを確認するには、次のコマンドを実行します。

```
cluster::> network connections active show -node [nodes] -service nfs* -remote-host [hostname]
```

たとえば、nconnectを使用しないアクティブなNFS接続からの出力を次に示します。

```
cluster::> network connections active show -node * -service nfs* -remote-host centos83-
perf.ntap.local
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                       Protocol/Service
----------- ---------------------- ------------------------------- ----------------
Node: node1
DEMO        data1:2049             centos83-perf.ntap.local:1013
                                                                   TCP/nfs
```

nconnectを使用している場合は、マウントあたりのCID数が増えます。この例では、nConnect=8を使用しています。

```
cluster::> network connections active show -node * -service nfs* -remote-host centos83-
perf.ntap.local
Vserver     Interface              Remote
Name        Name:Local Port        Host:Port                       Protocol/Service
----------- ---------------------- ------------------------------- ----------------
Node: node1
DEMO        data1:2049             centos83-perf.ntap.local:669 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:875 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:765 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:750 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:779 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:773 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:809 TCP/nfs
DEMO        data1:2049             centos83-perf.ntap.local:897 TCP/nfs
```

nconnectが使用されているかどうかを確認するもう1つの方法は、CIDオブジェクトの統計キャプチャを使用することです。次のコマンドを実行して、そのオブジェクトの統計を開始できます。

```
cluster::> set diag
cluster::*> statistics start -object cid
```

そのオブジェクトが実行されると、割り当てられたcids()の総数が追跡されますalloc_total。

たとえば、alloc_total nconnectを使用しないマウントのの数は次のとおりです。

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------------------- ----------------------------------------
    alloc_total                                                    11
```

これは、nconnect=4のマウントから取得したものです。

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------------------- ----------------------------------------
    alloc_total                                                    16
```

これは alloc_total [From nConnect=8]です。

```
cluster::*> statistics show -object cid -counter alloc_total

    Counter                                                     Value
    -------------------------------------------- ----------------------------------------
    alloc_total                                                    24
```

ハード/ソフト

ハードマウントオプションまたはソフトマウントオプションでは、NFSを使用するファイルを使用するプログラムを停止して、NFSサーバが使用できない場合にサーバがオンラインに戻るまで待機するか（hard）、エラーを報告するか（soft）を指定します。

hard を指定すると、この intr オプションも指定しないかぎり、使用できないNFSマウントに転送されたプロセスを終了できません。

softを指定した場合は、timeo=<value>オプションを指定できます。ここで、<value>はエラーが報告されるまでの秒数です。

注： ビジネスクリティカルなNFSエクスポートにはNetApp、ハードマウントの使用を推奨します。

intr / nointr

intrオプションを指定すると、マウントがハードマウントとして指定されている場合にNFSプロセスが中断されます。RHEL 6.4などの新しいクライアントではこのポリシーは廃止され、nointrにハードコードされています。Kill -9は、新しいカーネルでプロセスを中断する唯一の方法です。

注： ビジネスクリティカルなNFSエクスポートについてはNetApp、hardマウントとintrをサポートするNFSクライアントで使用することを推奨します。

ロギング、監視、統計

次のセクションでは、ログを表示する方法、NFS固有の環境の監視を設定する方法、およびNFS処理を管理する方法について説明します。

NFSロックの管理

ONTAP NFSサーバは、NASクライアントによってロックが確立されると、ロックを追跡します。ストレージ管理者は、必要に応じて、アドバンスト権限で次のコマンドを使用してロックを表示したり解除したりできます。

```
cluster::*> vserver locks
break show
```

さらに、NFSv4.xロックを指定するには、Diag権限で次のコマンドを実行します。

```
cluster::*> vserver locks nfsv4 show
```

NFSイベント:イベントメッセージングシステム

ONTAPは、クラスタ内のイベントを追跡するメッセージングシステムを提供します。これらは、すべてのサブシステムの情報、警告、およびエラー状態の概要を提供します。

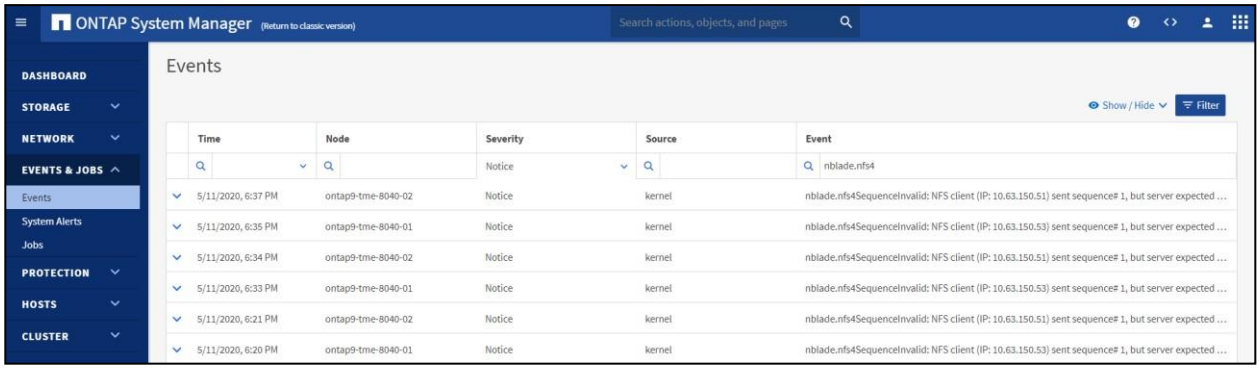
イベントメッセージングシステム（EMS）のログは、ONTAPシステムマネージャ（図17）を使用するか、CLIで次のコマンドを実行して表示できます。

```
cluster::> event log show
```

NFSイベントは、一般的なNFSイベント（エクスポートやその他のエラーなど）からネームサービスや認証の問題まで、さまざまなサブシステムを対象としています。CLIおよびGUIでは、イベントメッセージ名の一部でイベント名をフィルタリングできます。

```
cluster::*> event log show -message-name nblade.nfs*
Time                 Node             Severity      Event
-------------------- ---------------------- ---------------- ----------------------------------
5/11/2020 18:37:50 node2
                                      NOTICE        nblade.nfs4SequenceInvalid: NFS client (IP:
x.x.x.x) sent sequence# 1, but server expected sequence# 2. Server error: OLD_STATEID.
5/11/2020 18:35:52 node1
                                      NOTICE        nblade.nfs4SequenceInvalid: NFS client (IP:
x.x.x.y) sent sequence# 1, but server expected sequence# 2. Server error: OLD_STATEID.
5/11/2020 18:34:23 node2
                                      NOTICE        nblade.nfs4SequenceInvalid: NFS client (IP:
x.x.x.x) sent sequence# 1, but server expected sequence# 2. Server error: OLD_STATEID.
```

図17）ONTAP System Manager UIでのイベントのフィルタリング



EMSイベントは重大度レベルでトリガーされます。重大度レベルでは、どのメッセージが重要か（エラーや緊急など）、どのメッセージが単なる情報（情報や通知など）かを確認できます。一般に、デバッグレベルのメッセージは、環境で他に顕著な問題が発生していない限り無視してかまいません。エラー、アラート、緊急のメッセージが表示された場合は、サポートケースを開始する価値があります。

重大度、メッセージ名、およびその他の変数に基づいてフィルタリングできます。次のevent logコマンドを使用すると、NFS /マルチプロトコルNAS関連のEMSイベントを表示できます。

```
cluster::> event log show -message-name dns*
cluster::> event log show -message-name *export*
cluster::> event log show -message-name ldap*
cluster::> event log show -message-name mgmt.nfs*
cluster::> event log show -message-name nameserv*
cluster::> event log show -message-name nblade*
cluster::> event log show -message-name netgroup*
cluster::> event log show -message-name *nfs*
cluster::> event log show -message-name secd*
```

NFS統計

NFSの統計は、ONTAPカウンタマネージャのアーカイブ内に収集され、ONTAPシステムマネージャを使用して最大1年間表示できます。また、一般的なNFS統計のキャプチャは、Active IQ Performance Managerから実行できます。また、Grafana with NetApp Harvestなどのサードパーティのパフォーマンス監視ツールを使用することもできます。

NFSの個々のパフォーマンスカウンタが必要な場合は、 `statistics start` コマンドを使用して、特定のカウンタオブジェクトまたは複数のオブジェクトのキャプチャを有効にします。開始した統計カウンタは、ユーザが停止するまで実行されます。これらは、パフォーマンスが低下している可能性がある期間や、特定のボリュームのワークロードの傾向を確認する場合に最も便利です。

NFSでホストされているVMのタイプの確認

カウンタマネージャでキャプチャされた統計を使用して、ストレージにアクセスしているVMのタイプを確認できます。次のコマンドを実行して統計を表示します。これらは診断権限で使用できます。

```
cluster::> set diag
cluster::*> statistics start -object wafl -counter wafl_nfs_application_mask
cluster::*> statistics show -object wafl -counter wafl_nfs_application_mask -raw
```

これらの統計の出力には、特定のVMタイプのマスクが表示されます（表16）。

表16）VM統計マスク

VM タイプ	マスク
なし	0
ESX / ESXi	1
Citrix Xen	2
Red Hat KVM	4

複数のVMアプリケーションが使用されている場合は、マスクを追加して使用中のアプリケーションを特定します。たとえば、ESX / ESXiとRed Hat KVMが使用されている場合は、「1+4=5」となります。

Oracleを使用しているかどうかの確認

さらに、WAFLカウンタでは、OracleデータがNFSでホストされているかどうかを確認できます。

```
cluster::> set diag
cluster::*> statistics start -object wafl -counter wafl_nfs_oracle_wcount
cluster::*> statistics show -object wafl -counter wafl_nfs_oracle_wcount -raw
```

ONTAP 9.0以降のパフォーマンス監視の機能拡張

ONTAP 9.0以降では、ストレージ管理者がパフォーマンスの監視とプロアクティブなパフォーマンス管理を行う際に役立つ、パフォーマンス監視の機能拡張がいくつか導入されています。

注： パフォーマンスカウンタ、オブジェクト、およびインスタンスの完全なリストを表示するには、`statistics catalog` Advanced Privilegeにあるコマンドを実行してください。

上位のクライアント

ONTAPでは、上位クライアントと呼ばれる新しいONTAP 9機能を使用して、クラスタにデータを書き込む上位NASクライアントを追跡できます。この機能は受信処理全体を追跡し、クライアントIPアドレス、使用されているNASプロトコル、合計IOPS、アクセスされているノード、クライアントが接続しているSVMをリストします。

コマンドラインでadmin権限のコマンドを使用して、上位のクライアントをリアルタイムで監視することもできます `statistics top client show`。このコマンドを使用すると、最小30秒間隔、表示する反復回数、および表示するクライアントの最大数を指定できます。次の例では、2つのクライアントからNFSv3経由のNetApp FlexGroupボリュームに対してpythonファイル作成スクリプトを実行し、コマンドの出力例を示しています。

```
cluster::> statistics top client show -interval 30 -iterations 5 -max 10

cluster : 6/26/2017 17:42:27
*Estimated
     Total
      IOPS Protocol                Node Vserver Client
---------- ---------- ---------------------------- ----------------------------------------
     23010       nfs               node01 DEMO   x.x.x.b
     20006       nfs               node02 DEMO   x.x.x.c
        17     cifs                node02 CIFS
                                                  x.x.x.d
```

デフォルトでは、CLIはIOPSで順序付けされます。`-sort-key` オプションを使用してスループット別に注文することもできます。

例：

```
cluster::> statistics top client show -interval 30 -iterations 5 -max 10 -sort-key write_data

cluster : 6/26/2017 18:04:53
```

```
*Estimated
     Write
Data (Bps) Protocol                Node Vserver Client
---------- -------- ------------------------- ---------------------------------------------------------------
    154968      nfs               node01 DEMO   x.x.x.b
    150400      nfs               node02 DEMO   x.x.x.c
```

ホットファイル/上位ファイル

ONTAPでは、クラスタにアクセスする上位のクライアントを表示できるだけでなく、クラスタで使用されている上位のファイルも表示できます。これは、単一のESXサーバが、NFSでマウントされた単一のデータストア上で数百のVMをホストしているESX /仮想環境を扱う場合に特に役立ちます。その場合、上位のクライアント機能は、どのファイルが最も多くの作業を行っているかを知るほど有用ではありません。ONTAPは、その情報をVMDKレベルまで公開できます。

次の例では、このONTAPクラスタがNFSマウントされたデータストア上で複数のESXi VMをホストしています。この情報は、ONTAPシステムマネージャとCLIで確認できます。

CLIからは、`statistics top file show` 管理者権限でコマンドを実行できます。次の例では、Active IQ Unified Manager VMDKが書き込みスループットとIOPSの上位に使用されています。

```
cluster::> statistics top file show -interval 30 -iterations 1

cluster : 6/26/2017 18:01:21
*Estimated
     Total
      IOPS               Node Vserver     Volume File
---------- ---------------------- ---------- -------------- ------------------------------------------------------
        48             node03 vmware datastore1 /stme-ocum-01_1/stme-ocum-01_2-flat.vmdk
        31             node03 vmware datastore1 /OCUM722-MP/OCUM722-MP_2-flat.vmdk

cluster::> statistics top file show -interval 30 -iterations 1 -max 10 -sort-key write_data

cluster : 6/26/2017 18:05:04
*Estimated
     Write
Data (Bps)               Node Vserver     Volume File
---------- ---------------------- ---------- -------------- ---------------------------------------------------------------
    685600             node03 vmware datastore1 /Parisi OCUM 7.2/Parisi OCUM 7.2_2-flat.vmdk
    159475             node03 vmware datastore1 /stme-ocum-01_1/stme-ocum-01_2-flat.vmdk
```

注： 上記の例はラボ環境です。結果はワークロードによって異なります。

上位のファイルでサポートされるカウンタ

上位のファイル統計では、次のカウンタがサポートされています。

```
Object: top_file
    Counter                       Description
    ----------------------------------- ----------------------------------------------------------------
    other_ops                     Estimated Other Operations
    read_data                     Estimated Bytes From Read Operations
    read_ops                      Estimated Read Operations
    total_data                    Estimated Total Bytes From All Operations
    total_ops                     Estimated Total Operations
    write_data                    Estimated Bytes From Write Operations
    write_ops                     Estimated Write Operations
```

サポートされていないカウンタ（レイテンシなど）をで使用しようとすると `sort-key`、次のように表示されます。

```
cluster::*> statistics top file show -interval 30 -iterations 1 -max 10 -sort-key latency

Error: command failed: Unsupported sort-key counter: only "selector" counters are valid for
statistically tracked objects. For a list of valid sort-key counters, use this diagnostic
command: "statistics catalog counter show -object top_file -properties *selector*"
```

ストレージプール割り当ての表示

通常、パフォーマンスの問題がストレージプールのリソース不足によるものであると疑われる場合や、NFSv4マウントにアクセスできない場合は、NFSv4.xストレージプールの割り当ての表示が必要になることがあります。

ストレージプールの枯渇に関するその他の問題の例を次に示します。

- 1051413 - nblade NFSv4ストレージプールリソースの枯渇
- 1077736 - NFSv4トラフィックの実行中にNFS storepool string exhaustionが発生することがある
- 1158309 - storePool_StringAllocでNFSv4リソースが枯渇することがある
- 1312769 - StoreTypeSessionConnectionHolder storepool exhusts時にNFSv4.1のマウントが失敗する

ONTAP 8.2.3以降では、ストレージプールの枯渇が発生すると、EMSメッセージ（Nblade.nfsV4PoolExhaust）がトリガーされます。

storepoolオブジェクトの詳細については、以下を参照してください。

- storePoolオブジェクトとLock Managerオブジェクトの違い
- ストレージプールの枯渇が原因でNFSv4ファイルにアクセスできない

storepoolオブジェクトを表示するには、次のコマンドを実行します。

```
cluster::> set diag
cluster::*> statistics start -object nfsv4_diag
cluster::*> statistics show -object nfsv4_diag -counter *storePool_* -raw
```

問題のあるクライアントを特定するには、SSHを使用して次のコマンドをONTAPに送信します。

```
"set d -c off; rows 0; vserver locks nfsv4 show -inst; quit" | tee locks.txt | grep -i "client
name" | sort | uniq -c | sort -n"
```

問題でストレージプールの不足が疑われる場合は、NetAppサポートにお問い合わせください。一般に、NFSv4.xを使用する場合は、使用可能な最新のONTAPリリースを実行してください。

NFS使用状況の表示

ローカルノードで実行されたRPC呼び出しの数をNFSバージョンごとに確認できます。この値は保存され、ノードをリブートするまで消去されません。このコマンドにはdiagnostic権限レベルが必要で、ローカル ノードでのみ実行できます。

```
cluster::> set diag
cluster::*> system node nfs usage show
        Node: node2
                  v3: 120
                  v4: 2076
```

クラスタ内のアクティブなNFS接続の表示

ONTAPでは `network connections active show`、コマンドを使用して、クラスタ内のすべてのSVMとノードのアクティブなNFS接続を表示できます。このコマンドはIP、サービス、およびその他の項目でフィルタリングでき、より有用で詳細な情報を得られます。`netstat` 7-Modeで使用されていた従来のコマンドの代わりにコマンドを使用できます。

例：

```
cluster::> network connections active show
    show           show-clients   show-lifs      show-protocols show-services

cluster::> network connections active show -node node1 -service nfs*
             Vserver   Interface         Remote
      CID Ctx Name     Name:Local Port   Host:Port            Protocol/Service
```

```
------------ ----- ------------ ---------------------- -------------------------- ----------------------
Node: node1
286571835   6 vs0          data:2049         x.x.x.z:763    TCP/nfs

cluster::> network connections active show -node node2 -service nfs*
There are no entries matching your query.
```

また、を使用して、リスン状態のネットワーク接続を表示することもできます `network connections listening show`。

特定のボリュームへのNFSクライアントのマッピング

ONTAP 9.7では、ストレージ管理者がNFS経由でクラスタ内の特定のボリュームにマウントされているクライアントを確認できる新しいコマンドが導入されています。

ユースケースはさまざまですが、通常は次のいずれかのシナリオに該当します。

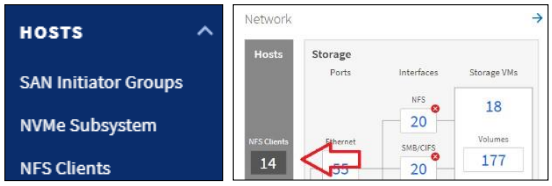
- 移行やカットオーバーなどを実行する前に、ボリュームを使用しているユーザを特定する必要がある
- 問題のトラブルシューティング
- 負荷の移動

接続されているNFSクライアントを表示するには：

```
cluster::> nfs connected-clients show ?
  [ -instance | -fields <fieldname>, ... ]
  [[-node] <nodename>]                                          Node Name
  [[-vserver] <vserver>]                                        Vserver
  [[-data-lif-ip] <IP Address>]                                 Data LIF IP Address
  [[-client-ip] <IP Address>]                                   Client IP Address
  [[-volume] <volume name>]                                     Volume Accessed
  [[-protocol] <Client Access Protocol>]                        Protocol Version
  [ -idle-time <[<integer>d][<integer>h][<integer>m][<integer>s]> ] Idle Time (Sec)
  [ -local-reqs <integer> ]                                     Number of Local Reqs
  [ -remote-reqs <integer> ]                                    Number of Remote Reqs
```

ONTAP 9.8では、これらのクライアントをONTAPシステムマネージャから表示できます。ダッシュボードで[NFS Clients]リンクをクリックするか、左側のメニューで[Hosts]→[NFS Clients]に移動します。

図18）ONTAP System ManagerでのNFSクライアントとボリュームのマッピングの表示



出力例については、「nfs connected-clientsの出力例」を参照してください。

NFSの高度な概念

このセクションでは、基本的な設定以外のNFSの概念について説明します。

umask

NFSではモード ビットを使用して権限を制御できます。モード ビットとは、数値によってファイルとフォルダの権限を決定する方法です。モード ビットの各数値は、読み取り、書き込み、実行、および特別な属性を決定します。各属性は次の数値で表されます。

- 実行=1

- 読み取り=2
- 書き込み=4

総合的な権限は、上記の数値を加算または減算することによって決まります。例：

```
4 + 2 + 1 = 7 (can do everything)
4 + 2 = 6 (rw) and so on…
```

UNIXアクセス権の詳細については、UNIXアクセス権のヘルプを参照してください。

umaskは、管理者がクライアントに許可する権限のレベルを制限できる機能です。デフォルトでは、ほとんどのクライアントのumaskは0022に設定されています。これは、そのクライアントから作成されたファイルにumaskが割り当てられることを意味します。umaskはオブジェクトの基本の権限から減算されます。権限が0777のボリュームを、NFSを使用してumaskが0022のクライアントにマウントすると、このクライアントからそのボリュームに書き込まれるオブジェクトの権限は0755（0777 - 0022）になります。

```
# umask
0022
# umask -S
u=rwx,g=rx,o=rx
```

ただし、多くのオペレーティング システムでは、実行権限を指定したファイルを作成することはできませんが、フォルダについては適切な権限を設定することが可能です。この場合、umask 0022で作成されたファイルの最終的な権限は0644になります。

次に、RHEL 6.5の使用例を示します。

```
# umask
0022
# cd /cdot
# mkdir umask_dir
# ls -la | grep umask_dir
drwxr-xr-x. 2 root      root         4096 Apr 23 14:39 umask_dir

# touch umask_file
# ls -la | grep umask_file
-rw-r--r--. 1 root      root            0 Apr 23 14:39 umask_file
```

NFSユーザnfsnobody

場合によっては、NFSクライアントのファイルリストにファイル所有者/グループ情報がと表示されることがあります nfsnobody。

```
# ls -la | grep newfile
-rwxrwxrwx   1 nfsnobody nfsnobody          0 May 19 13:30 newfile.txt
```

ファイルを数値で一覧表示すると、owner:groupがであることがわかります 65534。

```
# ls -lan | grep newfile
-rwxrwxrwx   1 65534 65534          0 May 19 13:30 newfile.txt
```

65534 ほとんどのLinuxクライアントではユーザがです nfsnobodyが、ONTAPではユーザがです pcuser。

```
cluster::*> unix-user show -vs rver DEMO -id 65534
                User            User    Group Full
Vserver         Name            ID      ID     Name
--------------- --------------- ------- ------ --------------------
DEMO            pcuser          65534 65534
```

anonymous エクスポートポリシールールのデフォルトユーザでもあります。

```
cluster::*> export-policy rule show -vserver DEMO -policyname default -fields anon
vserver policyname ruleindex anon
---------- ---------- ---------- --------
```

```
DEMO    default    1          65534
DEMO    default    2          65534
DEMO    default    3          65534
```

ONTAPクラスタのファイル権限を確認すると、UNIXの所有者がになっていることがわかります 65534が、WindowsのACLと所有者も異なります。

```
cluster::*> vserver security file-directory show -vserver DEMO -path /data/newfile.txt

               Vserver: DEMO
             File Path: /data/newfile.txt
     File Inode Number: 7088
        Security Style: ntfs
       Effective Style: ntfs
        DOS Attributes: 20
 DOS Attributes in Text: ---A----
Expanded Dos Attributes: -
          UNIX User Id: 65534
         UNIX Group Id: 65534
         UNIX Mode Bits: 777
 UNIX Mode Bits in Text: rwxrwxrwx
                   ACLs: NTFS Security Descriptor
                         Control:0x8004
                         Owner:NTAP\ntfs
                         Group:NTAP\DomainUsers
                         DACL - ACEs
                           ALLOW-Everyone-0x1f01ff-(Inherited)
```

`nfsnobody 65534` NFSのリストにまたはが表示される場合は、次のいずれかが発生している可能性が高くなります。

- NFSクライアントにエクスポートされているボリュームはWindows SMBクライアントでも使用され、共有に書き込むWindowsユーザは有効なUNIXユーザまたはグループにマッピングされません。
- NFSクライアントにエクスポートされているボリュームで匿名ユーザがに設定されている `65534` ため、NFSユーザが匿名ユーザに引き下げられています。引き下げの詳細については、「anonユーザ」を参照してください。

WindowsユーザとUNIXユーザのマッピングを表示するには、アドバンスト権限で次のコマンドを実行します。

```
cluster::*> access-check name-mapping show -vserver DEMO -direction win-unix -name ntfs
'ntfs' maps to 'pcuser'

cluster::*> access-check name-mapping show -vserver DEMO -direction win-unix -name prof1
'prof1' maps to 'prof1'
```

NFSv4.x：nobody：nobody

NFSv4.x構成で発生する最も一般的な問題の1つは、を使用してファイルまたはフォルダが `ls user:group` の組み合わせで所有されているとリストに表示される場合です `nobody:nobody`。

例：

```
sh-4.2$ ls -la | grep prof1-file
-rw-r--r-- 1 nobody nobody    0 Apr 24 13:25 prof1-file
```

数値IDはです `99`。

```
sh-4.2$ ls -lan | grep prof1-file
-rw-r--r-- 1 99 99    0 Apr 24 13:25 prof1-file
```

クラスタ（およびNFSv3を使用）では、そのファイル所有権に適切なUID / GIDが割り当てられているように見えます。

```
cluster::*> vserver security file-directory show -vserver DEMO -path /home/prof1/prof1-file
```

```
               Vserver: DEMO
             File Path: /home/prof1/prof1-file
     File Inode Number: 9996
        Security Style: unix
       Effective Style: unix
        DOS Attributes: 20
 DOS Attributes in Text: ---A----
Expanded Dos Attributes: -
          UNIX User Id: 1002
         UNIX Group Id: 10002
         UNIX Mode Bits: 644
 UNIX Mode Bits in Text: rw-r--r--
                   ACLs: -
```

場合によっては、ファイルに正しい所有者が表示されることがありますが、 `nobody` グループとして表示されることがあります。

```
sh-4.2$ ls -la | grep newfile1
-rw-r--r-- 1 prof1 nobody     0 Oct 9 2019 newfile1
```

誰が誰だ?

`nobody` NFSv4.xのユーザが `nfsnobody` 、「NFSユーザnfsnobody」で説明したユーザとは異なります。NFSクライアントが各ユーザをどのように認識しているかを表示するには、`id` コマンドを実行します。

```
# id nobody
uid=99(nobody) gid=99(nobody) groups=99(nobody)
# id nfsnobody
uid=65534(nfsnobody) gid=65534(nfsnobody) groups=65534(nfsnobody)
```

NFSv4.xでは `nobody idmapd.conf`、このユーザがファイルによって定義されたデフォルトユーザであり、使用する任意のユーザとして定義できます。

```
# cat /etc/idmapd.conf | grep nobody
#Nobody-User = nobody
#Nobody-Group = nobody
```

なぜこれが起こるのでしょうか?

名前文字列マッピングによるセキュリティはNFSv4.x処理の重要な原則であるため、名前文字列が適切に一致しない場合のデフォルトの動作では、ユーザとグループが所有するファイルやフォルダに通常アクセスできないユーザにそのユーザが引き下げられます。

`nobody` ファイルリストにユーザやグループが表示される場合、通常はNFSv4.xで何らかの設定が間違っていることを意味します。ここでは大文字と小文字の区別が重要になります。

たとえば 、user1@NTAP.LOCA L（uid 1234、gid 1234）がエクスポートにアクセスしている場合、ONTAP は user1@NTAP.LOCA L（uid 1234、gid 1234）を検索できる必要があります。ONTAPのユーザがUSER1@NTAP.LOCA Lの場合、 一致しません。多くの場合、クライアントのmessagesファイルには次の情報が表示されます。

```
May 19 13:14:29 centos7 nfsidmap[17481]: nss_getpwnam: name 'root@defaultv4iddomain.com' does not
map into domain 'NTAP.LOCAL'
May 19 13:15:05 centos7 nfsidmap[17534]: nss_getpwnam: name 'nobody' does not map into domain
'NTAP.LOCAL'
```

クライアントとサーバは、ユーザが実際にそのユーザであることに同意する必要があります。そのため、クライアントに表示されるユーザが、ONTAPに表示されるユーザと同じ情報を持っていることを確認するために、次の項目を確認する必要があります。

- NFSv4.x IDドメイン（クライアント：`idmapd.conf` ファイル、ONTAP： `-v4-id-domain` オプション）
- ユーザ名と数値ID（ネームサービススイッチ設定-クライアント： `nsswitch.conf` またはローカルのpasswdファイルとgroupファイル、ONTAP： `ns-switch` コマンド）

- グループ名と数値ID（ネームサービススイッチ構成-クライアント：nsswitch.conf ローカルのpasswdファイルとgroupファイル、ONTAP： ns-switch コマンド）

ほとんどの場合、 nobody クライアントのユーザおよびグループのリストにが表示されているにもかかわらず、ONTAPから正しいユーザおよびグループ情報が報告される場合（vserver security file-directory showを使用）、問題はユーザ名またはグループ名のドメインID変換です。

-v4-numeric-ids「名前文字列のバイパス–数値ID」で説明されているONTAPオプションを使用することもできます。

クライアントでのNFSv4.xの名前ID文字列の表示

NFSv4.xを使用している場合は、NFS処理中に名前文字列マッピングが実行されます。名前文字列が一致しないと、「NFSv4.x：nobody：nobody」のセクションで説明されている問題が表示されます。

を使用し /var/log/messages てNFSv4 IDの問題を検索するだけでなく、nfsidmap-l NFSクライアントでコマンドを使用して、NFSv4ドメインに適切にマッピングされているユーザ名を確認することもできます。

たとえば、クライアントとONTAP SVMの両方で存在するユーザがNFSv4.xマウントにアクセスしたあとのコマンドの出力例を次に示します。

```
# nfsidmap -l
4 .id_resolver keys found:
  gid:daemon@CENTOS-LDAP.LOCAL
  uid:nfs4@CENTOS-LDAP.LOCAL
  gid:root@CENTOS-LDAP.LOCAL
  uid:root@CENTOS-LDAP.LOCAL
```

NFSv4 IDドメインに適切にマッピングされていないユーザ（この場合は netapp-user）が同じマウントにアクセスしようとし、ファイルにタッチすると、 nobody:nobody想定どおりに割り当てられます。

```
# su netapp-user
sh-4.2$ id
uid=482600012(netapp-user), 2000(secondary)
sh-4.2$ cd /mnt/nfs4/
sh-4.2$ touch newfile
sh-4.2$ ls -la
total 16
drwxrwxrwx 5 root    root   4096 Jan 14 17:13 .
drwxr-xr-x. 8 root   root     81 Jan 14 10:02 ..
-rw-r--r--  1 nobody nobody    0 Jan 14 17:13 newfile
drwxrwxrwx 2 root    root   4096 Jan 13 13:20 qtree1
drwxrwxrwx 2 root    root   4096 Jan 13 13:13 qtree2
drwxr-xr-x 2 nfs4    daemon 4096 Jan 11 14:30 testdir
```

nfsidmap -l 出力にはユーザが pcuser 表示されますが、表示には表示されません。netapp-userこれはエクスポートポリシールール（65534）の匿名ユーザです。

```
# nfsidmap -l
6 .id_resolver keys found:
  gid:pcuser@CENTOS-LDAP.LOCAL
  uid:pcuser@CENTOS-LDAP.LOCAL
  gid:daemon@CENTOS-LDAP.LOCAL
  uid:nfs4@CENTOS-LDAP.LOCAL
  gid:root@CENTOS-LDAP.LOCAL
  uid:root@CENTOS-LDAP.LOCAL
```

Snapshotコピーの非表示

NetApp Snapshotコピーは、ボリュームの読み取り専用のポイントインタイム（PIT）イメージです。イメージにはSnapshotコピーが最後に作成されたあとに発生したファイルへの変更だけが記録されるため、ストレージ スペースは最小限しか消費せず、パフォーマンスのオーバーヘッドもわずかです。

ユーザはSnapshotコピーにアクセスして、NFSクライアントから個々のファイルをリストアできます。ただし、一部のワークロードでは、NFSマウントの内容がアプリケーションで表示されることがあります。場合によっては `.snapshot` ディレクトリが含まれるため、特にファイル数の多い環境では、これらのスキャンにかなりの時間がかかることがあります。その場合、NFSサーバオプションを使用して、Snapshotディレクトリをマウントで非表示にすることができます `-v3-hide-snapshot`。

コマンドで確認できるように、NFSv4.x `.snapshot` ではディレクトリがすでに非表示になっているため、NFSv3クライアントにのみ影響します。

`.snapshot` ディレクトリが非表示になっていても、アクセス可能です。`.snapshot` クライアントからディレクトリへのアクセスを削除するには、volumeオプションを使用し `-snapdir-access`ます。

注: これらのオプションは、エクスポートが再マウントされるまで有効になりません。

NFSクレデンシャルの表示と管理

ONTAP 9.3では、NASクレデンシャルとネームサービスサーバ管理のパフォーマンス、信頼性、耐障害性、サポート性を向上させるために、ネームサービス用のグローバルキャッシュが実装されました。

その一部として、NFSクレデンシャルキャッシュが実装されました。このキャッシュには、NFSエクスポートへのアクセス時にONTAPのユーザとグループの情報が格納されます。

これらのキャッシュは、Advanced Privilege `nfs credentials` コマンドを使用して表示および管理できます。

```
cluster::*> nfs credentials ?
  count                       *Count credentials cached by NFS
  flush                       *Flush credentials cached by NFS
  show                        *Show credentials cached by NFS
```

キャッシュエントリは、NFSマウントのTCP接続が存在するノードに入力されます。この情報は、クラスタで次のコマンドを実行すると確認できます。

```
cluster::*> nfs connected-clients show -vserver DEMO -client-ip x.x.x.x -fields data-lif-ip -
volume scripts
node                vserver data-lif-ip   client-ip     volume protocol
------------------- ------- ------------- ------------- ------- ---------
Node1               DEMO    x.x.x.y       x.x.x.x       scripts nfs3
```

上記のコマンドから、クライアントIP `x.x.x.x` がnode1のデータLIFに接続されていることがわかります。これにより、キャッシュエントリに焦点を当てるノードを絞り込むことができます。

`nfs credentials count` コマンドを使用すると、NFSクレデンシャルキャッシュに現在格納されているクレデンシャルの数を確認できます。これは、キャッシュをクリアした場合の影響を理解するのに役立ちます。

```
cluster::*> nfs credentials count -node node1
Number of credentials cached by NFS on node "node1": 4
```

ユーザがONTAP NFSエクスポートに移動すると、ユーザID、グループIDなどがすべてNFSクレデンシャルキャッシュに追加されます。たとえば、という名前のユーザがい `prof1`ます。

```
# id prof1
uid=1102(prof1) gid=10002(ProfGroup) groups=10002(ProfGroup),10000(Domain
Users),1202(group2),1101(group1),1220(sharedgroup),1203(group3)
```

このユーザには8つの異なるエントリがあります。数字のUIDと7つのグループメンバーシップです。次に、ユーザは `prof1` NFSエクスポートにアクセスします。クレデンシャルキャッシュが8倍になります。

```
cluster::*> nfs credentials count -node node1
Number of credentials cached by NFS on node "node1": 12
```

この数は、SVM単位だけでなく、ノード全体に対してもカウントされます。環境内に複数のSVMがある場合は、トラブルシューティングの際にこの数を使用しないことがあります。

NFSクレデンシャルキャッシュの表示

NFSクレデンシャルキャッシュに格納されているクレデンシャルの数だけでなく、ユーザやグループの個々のキャッシュエントリも表示できます。環境内のユーザからアクセスの問題について苦情があった場合は、キャッシュ内でそのユーザを検索できます。

注： クレデンシャルキャッシュ全体の内容を表示することはできません。

この例では、が `prof1` マウントにアクセスしています。キャッシュエントリと、キャッシュエントリに関する詳細を示すフラグが表示されます。

```
cluster::*> nfs credentials show -node node1 -vserver DEMO -unix-user-name prof1

Credentials
---------------
                      Node: node1
                   Vserver: DEMO
                 Client IP: -
                     Flags: unix-extended-creds-present, id-name-mapping-present
  Time since Last Refresh: 52s
   Time since Last Access: 44s
                 Hit Count: 4

  UNIX Credentials:
                     Flags: 1
                 Domain ID: 0
                       UID: 1102
               Primary GID: 10002
           Additional GIDs: 10002
                            10000
                            1101
                            1202
                            1203
                            1220

  Windows Credentials:
                     Flags: -
                  User SID: -
         Primary Group SID: -
               Domain SIDs: -

  ID-Name Information:
                      Type: user
                        ID: 1102
                      Name: prof1
```

ユーザのプライマリグループのエントリを表示することもできます。

```
cluster::*> nfs credentials show -node node1 -vserver DEMO -unix-group-name ProfGroup

Credentials
---------------
                      Node: node1
                   Vserver: DEMO
                 Client IP: -
                     Flags: id-name-mapping-present
  Time since Last Refresh: 64s
   Time since Last Access: 6s
                 Hit Count: 2

  UNIX Credentials:
                     Flags: -
                 Domain ID: -
                       UID: -
               Primary GID: -
           Additional GIDs: -

  Windows Credentials:
                     Flags: -
```

```
                User SID: -
        Primary Group SID: -
              Domain SIDs: -

  ID-Name Information:
                    Type: group
                      ID: 10002
                    Name: ProfGroup
```

また、アクセスを試行したクライアントIPまで、ユーザとグループのクレデンシャルキャッシュエントリを表示することもできます。

```
cluster::*> nfs credentials show -node node1 -vserver DEMO -client-ip x.x.x.x -unix-user-id 1102

Credentials
---------------
                         Node: node1
                      Vserver: DEMO
                    Client IP: x.x.x.x
                        Flags: unix-extended-creds-present, id-name-mapping-present
       Time since Last Refresh: 35s
        Time since Last Access: 34s
                    Hit Count: 2
               Reference Count: 4 Result
 of Last Update Attempt: no error

  UNIX Credentials:
                     Flags: 1
                 Domain ID: 0
                       UID: 1102
               Primary GID: 10002
          Additional GIDs: 10002
                           10000
                           1101
                           1202
                           1203
                           1220

  Windows Credentials:
                   Flags: -
                User SID: -
        Primary Group SID: -
              Domain SIDs: -

  ID-Name Information:
                    Type: user
                      ID: 1102
                    Name: prof1
```

クレデンシャルキャッシュでは、負のエントリ（解決できなかったエントリ）もキャッシュに保持されます。負のエントリは、ONTAPが数値のUIDを有効なユーザに解決できない場合に発生します。この場合、UID 1236はONTAPで解決できませんが、NFSエクスポートへのアクセスが試行されます。

```
# su cifsuser
bash-4.2$ cd /scripts/
bash: cd: /scripts/: Permission denied
bash-4.2$ id
uid=1236(cifsuser) gid=1236(cifsuser) groups=1236(cifsuser)

cluster::*> nfs credentials show -node node1 -vserver DEMO -unix-user-id 1236

Credentials
---------------
                     Node: node1
                  Vserver: DEMO
                Client IP: -
                    Flags: no-unix-extended-creds, no-id-name-mapping
  Time since Last Refresh: 33s
   Time since Last Access: 7s
```

```
               Hit Count: 15

  UNIX Credentials:
                    Flags: -
                Domain ID: -
                      UID: -
              Primary GID: -
           Additional GIDs: -

  Windows Credentials:
                    Flags: -
                 User SID: -
        Primary Group SID: -
              Domain SIDs: -

  ID-Name Information:
                     Type: -
                       ID: -
                     Name: -
```

NFSv4.xおよびマルチプロトコルNASでのNFSクレデンシャルキャッシュ

NFSクレデンシャルキャッシュエントリには、WindowsクレデンシャルとNFSv4 IDマッピングクレデンシャルも格納されます。

ユーザがNFSv4.xエクスポートを通過してIDドメインに正しくマッピングされると、`ID-Name Information` フィールドに値が入力されていることがわかります。

```
Credentials
---------------
                          Node: node
                        Vserver: DEMO
                      Client IP: x.x.x.x
                          Flags: unix-extended-creds-present, id-name-mapping-present
       Time since Last Refresh: 12s
        Time since Last Access: 9s
                     Hit Count: 2
                Reference Count: 4 Result
 of Last Update Attempt: no error

  UNIX Credentials:
                    Flags: 1
                Domain ID: 0
                      UID: 1102
              Primary GID: 10002
           Additional GIDs: 10002
                            10000
                            1101
                            1202
                            1203
                            1220

  Windows Credentials:
                    Flags: -
                 User SID: -
        Primary Group SID: -
              Domain SIDs: -

  ID-Name Information:
                     Type: user
                       ID: 1102
                     Name: prof1
```

ユーザがNTFS権限/セキュリティ形式のエクスポートにアクセスすると、フラグ `cifs-creds- present`とドメインSID情報が `Windows Credentials`次の場所に表示されます。

```
---------------
Credentials

                          Node: node1
                        Vserver: DEMO
```

```
                    Client IP: x.x.x.x
                        Flags: ip-qualifier-configured, unix-extended-creds-present, cifs-creds-
present
       Time since Last Refresh: 19s
        Time since Last Access: 1s
                     Hit Count: 9
                Reference Count: 2 Result
 of Last Update Attempt: no error

  UNIX Credentials:
                       Flags: 0
                   Domain ID: 0
                         UID: 1102
                Primary GID: 10002
           Additional GIDs: 10002
                            10000
                            1101
                            1202
                            1203
                            1220

  Windows Credentials:
                       Flags: 8320
                  User SID: S-1-5-21-3552729481-4032800560-2279794651-1214
         Primary Group SID: S-1-5-21-3552729481-4032800560-2279794651-513
                Domain SIDs: S-1-5-21-3552729481-4032800560-2279794651
                            S-1-18
                            S-1-1
                            S-1-5
                            S-1-5-32

  ID-Name Information:
                       Type: -
                         ID: -
                       Name: -
```

NFSクレデンシャルキャッシュノセッテイ

NFSクレデンシャルキャッシュのタイムアウト値は、表17に示すNFSサーバオプションによって制御されます。

表17）NFSクレデンシャルキャッシュの設定

オプション	キノウ	デフォルト値（ms）
-cached-cred-negative-ttl	このパラメータはオプションで、ネガティブ キャッシュされたクレデンシャルがキャッシュからクリアされるまでの時間を指定します。60000 ~ 604800000の値を指定する必要があります。	7200000（2時間）
-cached-cred-positive-ttl	このパラメータはオプションで、ポジティブ キャッシュされたクレデンシャルがキャッシュからクリアされるまでの時間を指定します。60000 ~ 604800000の値を指定する必要があります。	86400000（24時間）
-cached-cred-harvest-timeout	（オプション）このパラメータは、キャッシュされたクレデンシャルの収集タイムアウトを指定します60000 ~ 604800000の値を指定する必要があります。	86400000（24時間）

キャッシュエントリには、最終アクセス/更新からの時間が保持されます（ show コマンドを参照）。エントリが一定期間アイドル状態のままになると、最終的にはキャッシュから削除されます。エントリがアクティブな場合は、エントリが更新されてキャッシュに残ります。

これらの値は、必要な影響に応じて、タイムアウト値を長くしたり短くしたりできます。

- **キャッシュタイムアウト値を長くする** と、ネットワークの負荷が軽減され、ユーザの検索が高速になりますが、キャッシュエントリがネームサービスと常に同期されているとは限らないため、誤検出や誤検出が増加する可能性があります。
- **キャッシュタイムアウト値を短く** すると、ネットワークとネームサーバの負荷が増大し、（ネームサービスソースによっては）ネーム検索のレイテンシが増大する可能性がありますが、エントリの正確性と最新性は向上します。

値はそのままにしておくことを推奨します。値を変更する必要がある場合は、結果を確認し、必要に応じて調整してください。

NFSクレデンシャルキャッシュのフラッシュ

ユーザがグループに対して追加または削除され、適切なアクセス権がない場合は、キャッシュエントリがタイムアウトするのを待たずに、クレデンシャルキャッシュエントリを手動でフラッシュできます。

このコマンドは、UNIXユーザ、数値ID、UNIXグループ、数値IDに対して実行できます。また、問題を持つクライアントIPアドレスまで、コマンドをきめ細かく実行できます。

```
cluster::*> nfs credentials flush -node node1 -vserver DEMO -client-ip x.x.x.x -unix-user-id 1102

Number of matching credentials flushed: 2
```

注： フラッシュできるNFSクレデンシャルキャッシュエントリは一度に1つだけです。

NFSクレデンシャルキャッシュはネームサービスキャッシュとは別のキャッシュです。ネームサービスキャッシュの管理については、TR-4835：『How to Configure LDAP in ONTAP』を参照してください。

NFSv3マウントでのNFSv4.x ACLの使用

デフォルトでは、NFSv3では権限の管理方法がかなり限定されています。ただし、ONTAPでは、ファイルやフォルダにNFSv4.x ACLを設定して、NFSv3エクスポートに適用することができます。

これを行う方法は簡単です。

1. ONTAPおよびクライアントでNFSv4.xを設定して有効にします。
2. ONTAPでNFSv4.x ACLのサポートを有効にします。
3. エクスポートをNFSv4.xでマウントします。
4. NFSv4.x ACLを適用します。
5. NFSv3を使用してエクスポートをアンマウントして再マウントし、テストします。

この例の環境では、homedirボリュームをマウントし、 `prof1` を使用してというユーザのrootが所有するファイルにACLを設定しまし `nfs4_setfacl -e` た（これにより、長いコマンドを入力する必要がなく、ファイルを編集できます）。

ファイルはrootユーザのhomedirにあります。ルートhomedirは755に設定されています。これは、誰でもそれらを読み取ることができますが、所有者（ルート）以外の誰もそれらに書き込むことができないことを意味します。

```
drwxr-xr-x 2 root root 4096 Jul 13 10:42 root
```

つまり、ユーザにフルコントロールを許可するようにNFSv4.x ACLを設定しないかぎり、

```
[root@centos7 mnt]# nfs4_getfacl /mnt/root/file
A::prof1@ntap.local:rwaxtTnNcCy
A::OWNER@:rwaxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy
```

これらの権限は、ONTAP CLIからも確認できます。

```
cluster::*> vserver security file-directory show -vserver DEMO -path /home/root/file
```

```
Vserver: DEMO
 File Path: /home/root/file
 File Inode Number: 8644
 Security Style: unix
 Effective Style: unix
 DOS Attributes: 20
 DOS Attributes in Text: ---A----
Expanded Dos Attributes: -
 UNIX User Id: 0
 UNIX Group Id: 1
 UNIX Mode Bits: 755
 UNIX Mode Bits in Text: rwxr-xr-x
 ACLs: NFSV4 Security Descriptor
 Control:0x8014
 DACL - ACEs
 ALLOW-user-prof1-0x1601bf
 ALLOW-OWNER@-0x1601bf
 ALLOW-GROUP@-0x1200a9-IG
 ALLOW-EVERYONE@-0x1200a9
```

上記の例では prof1、ファイルの完全な制御を提供しています。その後、NFSv3を使用してマウントしました。NFSv4.x ACLに関連付けられていないユーザになった場合、ファイルへの書き込みや削除はできません（想定される動作）。

```
[root@centos7 /]# su student1
sh-4.2$ cd /mnt/root
sh-4.2$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Jul 13 10:42 .
drwxrwxrwx 11 root root 4096 Jul 10 10:04 ..
-rwxr-xr-x 1 root bin 0 Jul 13 10:23 file
-rwxr-xr-x 1 root root 0 Mar 29 11:37 test.txt

sh-4.2$ touch file
touch: cannot touch 'file': Permission denied
sh-4.2$ rm file
rm: remove write-protected regular empty file 'file'? y
rm: cannot remove 'file': Permission denied
```

prof1 ユーザを変更すると、v3のモードビット権限では実行できないように指定されていても、必要な操作を実行できるようになります。NFSv4.x ACLが機能しているためです。

```
[root@centos7 /]# su prof1
sh-4.2$ cd /mnt/root
sh-4.2$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Jul 13 10:42 .
drwxrwxrwx 11 root root 4096 Jul 10 10:04 ..
-rwxr-xr-x 1 root bin 0 Jul 13 10:23 file
-rwxr-xr-x 1 root root 0 Mar 29 11:37 test.txt

sh-4.2$ vi file
sh-4.2$ cat file
NFSv4ACLS!
```

chmodただし、を実行しても、ユーザのNFSv4 ACLからは何も変更されていないようです。ファイルに700を設定し、NFSv3モードビットで表示されました。

```
sh-4.2$ chmod 700 file
sh-4.2$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Jul 13 10:42 .
drwxrwxrwx 11 root root 4096 Jul 10 10:04 ..
-rwx------ 1 root bin 11 Aug 11 09:58 file
-rwxr-xr-x 1 root root 0 Mar 29 11:37 test.txt
```

しかし、 prof1 ユーザーがまだ完全な制御を持っていることに注意してください。

```
cluster::*> vserver security file-directory show -vserver DEMO -path /home/root/file

Vserver: DEMO
 File Path: /home/root/file
 File Inode Number: 8644
 Security Style: unix
 Effective Style: unix
 DOS Attributes: 20
 DOS Attributes in Text: ---A----
Expanded Dos Attributes: -
 UNIX User Id: 0
 UNIX Group Id: 1
 UNIX Mode Bits: 700
 UNIX Mode Bits in Text: rwx------
 ACLs: NFSV4 Security Descriptor
 Control:0x8014
 DACL - ACEs
 ALLOW-user-prof1-0x1601bf
 ALLOW-OWNER@-0x1601bf
 ALLOW-GROUP@-0x120088-IG
 ALLOW-EVERYONE@-0x120088
```

NFSv4.x ACL保持が有効になっているためです。このオプションをディセーブルにすると、 `chmod` ACLが消去されます。

権限の問題をトラブルシューティングするためのコマンド

ほとんどの場合、NFS権限の問題はきわめて簡単です。NFSv3では基本的なRWXモードビットが使用されます。ただし、NFSv4 ACLやマルチプロトコルNASアクセス、およびさまざまなセキュリティ形式が関係すると、状況が複雑になる可能性があります。このセクションでは、NAS環境での権限の問題のトラブルシューティングに役立つコマンドをいくつか紹介します。ネームサービスキャッシュの情報については、「NFSクレデンシャルの表示と管理」を参照してください。詳細については、TR-4835：『LDAP in NetApp ONTAP』を参照してください。

UNIX UIDおよびグループメンバーシップの確認

NFSv3処理では、IDを確認するために数値を渡すことができるため、UNIXのユーザ名とグループ名はそれほど重要ではありません。ただし、NFSv4およびNTFSセキュリティ形式のオブジェクトでは、適切な名前解決のために、数値IDを有効なUNIXユーザ名およびグループ名に変換する必要があります。NFSv4の場合は、nobodyにユーザが引き下げられないようにするために必要です。NTFSセキュリティ形式では、UNIXユーザ名を有効なWindowsユーザ名にマッピングする必要があります。

ONTAPには、UNIXユーザのIDとグループメンバーシップを表示するためのコマンドがいくつかあります。ローカルUNIXユーザおよびグループの場合は、次のコマンドを実行します。

```
cluster::> unix-user show
cluster::> unix-group show
```

すべてのUNIXユーザ（ローカルおよびネームサービス、高度な権限）のUID / GIDの基本情報を表示するには、次のコマンドを実行します。

```
cluster::*> access-check authentication show-ontap-admin-unix-creds
```

または

```
cluster::*> getxxbyyy getpwbyname -node node1 -vserver DEMO -user name prof1 -show-source true
  (vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: LDAP
pw_name: prof1
pw_passwd:
pw_uid: 1102
pw_gid: 10002
pw_gecos:
pw_dir:
pw_shell:
```

```
cluster::*> getxxbyyy getpwbyname -node node1 -vserver DEMO -user name host -show-source true
  (vserver services name-service getxxbyyy getpwbyname)
Source used for lookup: Files
pw_name: host
pw_passwd: *
pw_uid: 598
pw_gid: 0
pw_gecos:
pw_dir:
pw_shell:
```

ユーザ情報とグループメンバーシップ（ローカルサービスとネームサービス、高度な権限）を表示するには、次のコマンドを実行します。

```
cluster::*> getxxbyyy getgrlist -node node1 -vserver DEMO -user name prof1
  (vserver services name-service getxxbyyy getgrlist)
pw_name: prof1
Groups: 10002 10002 10000 1101 1202 1203 48
```

マルチプロトコルユーザのユーザおよびグループ情報の表示

CIFS / SMBとNFSの両方が設定されている環境では、ユーザ名、ネームマッピング、ID、グループ名、権限、およびグループメンバーシップは、Advanced Privilegeの1つのコマンドで実行できます。このコマンドは、マルチプロトコル環境で使用する場合に推奨されるコマンドです。コマンドは、SMB / CIFSサーバが設定されていない場合は機能しません。

```
cluster::*> access-check authentication show-creds -node node1 -vserver DEMO -unix-user-name
prof1 -list-name true -list-id true
  (vserver services access-check authentication show-creds)

 UNIX UID: 1102 (prof1) <> Windows User: S-1-5-21-3552729481-4032800560-2279794651-1110
(NTAP\prof1 (Windows Domain User))

 GID: 10002 (ProfGroup)
 Supplementary GIDs:
  10002 (ProfGroup)
  10000 (Domain Users)
  1101 (group1)
  1202 (group2)
  1203 (group3)
  48 (apache-group)

 Primary Group SID: S-1-5-21-3552729481-4032800560-2279794651-1111   NTAP\ProfGroup (Windows
Domain group)

 Windows Membership:
  S-1-5-21-3552729481-4032800560-2279794651-1301   NTAP\apache-group (Windows Domain group)
  S-1-5-21-3552729481-4032800560-2279794651-1106   NTAP\group2 (Windows Domain group)
  S-1-5-21-3552729481-4032800560-2279794651-513   NTAP\DomainUsers (Windows Domain group)
  S-1-5-21-3552729481-4032800560-2279794651-1105   NTAP\group1 (Windows Domain group)
  S-1-5-21-3552729481-4032800560-2279794651-1107   NTAP\group3 (Windows Domain group)
  S-1-5-21-3552729481-4032800560-2279794651-1111 NTAP\ProfGroup (Windows Domain group) S-
  1-5-21-3552729481-4032800560-2279794651-1231 NTAP\local-group.ntap (Windows Alias) S-
  1-18-2 Service asserted identity (Windows Well known group)
  S-1-5-32-551   BUILTIN\Backup Operators (Windows Alias)
  S-1-5-32-544   BUILTIN\Administrators (Windows Alias)
  S-1-5-32-545   BUILTIN\Users (Windows Alias)
 User is also a member of Everyone, Authenticated Users, and Network Users

 Privileges (0x22b7):
  SeBackupPrivilege
  SeRestorePrivilege
  SeTakeOwnershipPrivilege
  SeSecurityPrivilege
  SeChangeNotifyPrivilege
```

ONTAPテシヨウサレルフアイルケンケンノヒヨウシ

権限の問題をトラブルシューティングする際に、NASクライアントから権限を表示するアクセス権がない場合があります。また、NASクライアントに表示されている権限とONTAPに表示されている権限を確認することもできます。これを行うには、次のコマンドを実行します。

```
cluster::> file-directory show -vserver DEMO -path /home/prof1
  (vserver security file-directory show)

               Vserver: DEMO
             File Path: /home/prof1
     File Inode Number: 8638
        Security Style: ntfs
       Effective Style: ntfs
        DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
          UNIX User Id: 0
         UNIX Group Id: 0
        UNIX Mode Bits: 777
 UNIX Mode Bits in Text: rwxrwxrwx
                   ACLs: NTFS Security Descriptor
                         Control:0x8504
                         Owner:NTAP\prof1
                         Group:BUILTIN\Administrators
                         DACL - ACEs
                           ALLOW-Everyone-0x1f01ff-OI|CI
                           ALLOW-NTAP\prof1-0x1f01ff-OI|CI
                           ALLOW-NTAP\sharedgroup-0x1200a9-OI|CI
                           ALLOW-NTAP\Administrator-0x1f01ff-OI|CI
```

また、特定のユーザが特定のファイルまたはディレクトリに対して有効になっている権限を確認することもできます。

```
cluster::> file-directory show-effective-permissions -vserver DEMO -unix-user-name prof1 -path
/home/prof1
  (vserver security file-directory show-effective-permissions)

                       Vserver: DEMO
             Windows User Name: NTAP\prof1
                Unix User Name: prof1
                     File Path: /home/prof1
               CIFS Share Path: -
         Effective Permissions:
                                Effective File or Directory Permission: 0x1f01ff
                                       Read
                                       Write
                                       Append
                                       Read EA
                                       Write EA
                                       Execute
                                       Delete Child
                                       Read Attributes
                                       Write Attributes
                                       Delete
                                       Read Control
                                       Write DAC
                                       Write Owner
                                       Synchronize
```

エクスポートポリシーアクセスの確認

場合によっては、エクスポートポリシーの設定が原因で権限の問題が発生することがあります。たとえば、読み取りのみを許可するようにポリシーが設定されている場合は、マウントで設定されているすべてのユーザ権限が上書きされる可能性があります。

ONTAPでは、次のコマンドを実行して、クライアントのエクスポートポリシーアクセスをチェックできます。

```
cluster::> export-policy check-access
```

セキュリティトレースの使用

権限の問題が発生したときにトレースする場合は、セキュリティトレースフィルタ機能を使用して、NFS権限とSMB / CIFS権限の両方をトレースできます。

トレースフィルタを作成するには、次のコマンドを実行します。

```
cluster::> vserver security trace filter create ?
    -vserver <vserver name>         Vserver
   [-index] <integer>               Filter Index
  [[-protocols] {cifs|nfs}, ...]    Protocols (default: cifs)
  [ -client-ip <IP Address> ]       Client IP Address to Match
  [ -path <TextNoCase> ]            Path
  { [ -windows-name <TextNoCase> ] Windows User Name
  | [ -unix-name <TextNoCase> ] }   UNIX User Name or User ID
  [ -trace-allow {yes|no} ]         Trace Allow Events (default: no)
  [ -enabled {enabled|disabled} ]   Filter Enabled (default: enabled)
  [ -time-enabled {1..720} ]        Minutes Filter is Enabled (default: 60)
```

必要に応じて、特定のユーザ名またはIPアドレスにトレースを絞り込むことができます。

```
cluster::*> vserver security trace filter modify -vserver DEMO -index 1 -protocols nfs -client-ip
x.x.x.x -trace-allow yes -enabled enabled
```

トレースが作成されると、結果がリアルタイムで表示されます。結果を表示するときに、成功、失敗、ユーザID、プロトコルなどでフィルタリングできます。

```
cluster::> vserver security trace trace-result show ?
  [ -instance | -fields <fieldname>, ... ]
  [[-node] <nodename>]                  Node
  [ -vserver <vserver name> ]           Vserver
  [[-seqnum] <integer>]                 Sequence Number
  [ -keytime <Date> ]                   Time
  [ -index <integer> ]                  Index of the Filter
  [ -client-ip <IP Address> ]           Client IP Address
  [ -path <TextNoCase> ]                Path of the File Being Accessed
  [ -win-user <TextNoCase> ]            Windows User Name
  [ -security-style <security style> ] Effective Security Style On File
  [ -result <TextNoCase> ]              Result of Security Checks
  [ -unix-user <TextNoCase> ]           UNIX User Name
  [ -session-id <integer> ]             CIFS Session ID
  [ -share-name <TextNoCase> ]          Accessed CIFS Share Name
  [ -protocol {cifs|nfs} ]              Protocol
  [ -volume-name <TextNoCase> ]         Accessed Volume Name
```

特定のユーザに対する権限/アクセスエラーの例を次に示します。

```
cluster::> vserver security trace trace-result show -node * -vserver DEMO -unix-user 1102 -result
*denied*

Vserver: DEMO


Node             Index Filter Details              Reason
---------------- ----- ----------------------------------- ----------------------------------------
Node2            1     Security Style: UNIX and    Access is denied. The
                       NFSv4 ACL                   requested permissions are not
                                                   granted by the ACE while
                                                   setting attributes. Access is
                                                   not granted for: "Write DAC"
                       Protocol: nfs
                       Volume: home
                       Share: -
                       Path: /dir
                       Win-User: -
```

```
                    UNIX-User: 1102
                    Session-ID: -
```

動的NAS TCP自動チューニング

ONTAPは、NASプロトコルスタックが最適な設定にオンザフライでバッファサイズを調整できる動的なNAS TCP自動チューニングを使用します。これは、静的な方法でのTCPバッファ サイズの設定では、常に変動するネットワークの状況にも、システムに対して同時に発生するさまざまな種類の接続にも対応できないためです。オートチューニングは、アプリケーションのデータ読み取り速度とシステムのデータ受信速度を計算して最適なバッファ サイズを割り出すことによって、NAS TCP接続のスループットを最適化します。この機能はあらかじめ設定されており変更はできません。また、バッファ サイズは増えるだけで、減ることはありません。バッファ サイズの初期値は32Kです。オートチューニングは、グローバルに適用されるのではなく、個々のTCP接続に適用されます。

これは`tcp-max-xfer-size`、NFSサーバオプションの最大転送サイズの値（-）とは異なります。

ONTAP 9以降での最大転送サイズの設定

NetApp ONTAP 9以降では `-v3-tcp-max-read-size` `-v3-tcp-max-write-size` 廃止されています。NetAppで `-tcp-max-xfer-size` は、代わりにオプションを使用することを推奨します。この変更により、読み取りと書き込みの両方で1MBのTCP転送サイズも許可されます。ONTAP 9より前のバージョンのONTAPでは、読み取りに1MBしか使用できませんでした。

注：これらの値を調整すると、新しいマウントにのみ影響します。既存のマウントは、マウント時に設定されたブロック サイズを維持します。サイズを変更すると、既存のマウントで書き込み処理が拒否されたり、読み取りに対する応答が要求よりも小さくなったりすることがあります。ブロックサイズオプションを変更する場合は、必ずクライアントをアンマウントして再マウントし、変更を反映してください。詳細については、バグ962596を参照してください。

動的なウィンドウサイズを選択する理由

ウィンドウ サイズは一般に個々のホストまたは接続を対象に考慮されるため、ほとんどの環境ではTCPウィンドウ サイズを固定することにメリットはありません。ONTAPで実行されるNFSなどのサーバでは、複数のホストへの接続が複数存在します。各接続にはそれぞれ特徴があり、必要とされるスループットもさまざまです。静的なウィンドウでは、サーバは多様なインバウンド接続を適切に処理することができません。ネットワーク インフラを構成する要素は多くの場合変動し、静的であることはまれです。このため、TCPスタックはそれらの通信を効率的かつ効果的に処理できる必要があります。動的なウィンドウ サイズを使用すると、静的ウィンドウ環境での問題（ネットワークの利用率が高すぎるとスループットが低下する、あるいはネットワークの利用率が低すぎると運用効率が徐々に低下するなど）を回避できます。

EXECコンテキストスロットリング

クライアントからNFS処理が送信されると、ONTAPはノード上のリソースをプレースホルダとして予約します。この操作は実行コンテキスト（EXECコンテキスト）と呼ばれます。操作が完了すると、EXECコンテキストはシステムに解放されて使用されます。

これらのリソースはノード内に限りがあり、ONTAPのバージョンやプラットフォーム/メモリなどの要素に依存します。これらの値のノード単位の制限は、次のコマンドを実行して確認できます。

```
cluster::> set diag
cluster::*> systemshell -node node1 -command "sysctl -a | grep preallocated"
```

表18 に、使用可能なEXECコンテキストのノード単位の値の例を示します。

表18）ノードごとのExecコンテキスト

ノード タイプ	ONTAPのバージョン	事前割り当て済みEXECコンテキストの総数
AFF8040	9.8	1500
AFF A300	9.9.1	3000
AFF A800	9.8	10000
FAS9000	9.9.1	10000

ノード単位の制限に加えて、TCP CIDごとに128の同時操作（割り当てられたEXECコンテキスト）の制限もあります。クライアントから128を超える同時処理が送信された場合、ONTAPは新しいリソースが解放されるまでそれらの処理をブロックします。デフォルトでは、Linuxクライアントはこれらの処理のうち最大65,536個を同時に送信するように設定されているため、比較的短時間で制限に到達できます。

このプロセスの意味については、「RPCスロットテーブルの潜在的な問題の特定」で詳しく説明しています。

場合によっては、1つのクライアントがノードのWAFLレイヤに同じボリュームへの要求を大量に送信し、その結果WAFLのレイテンシが増大し、EXECコンテキストが解放されてシステムに解放されるまでの時間が長くなることがあります。これにより、同じノードに接続している他のワークロードで使用できるEXECコンテキストの総数が削減されます。これは、多数のクライアントが一度に同じボリュームにNFS処理を送信するグリッドコンピューティングアプリケーションでよく見られます。たとえば、すべてのNFSクライアントが同時に128の処理を送信する場合、その制限を超えるまでには24クライアントしか必要ありません。

ONTAP 9.9.1では、RAMが256GBを超えるプラットフォーム向けの新機能が導入されています。この機能は、BullyワークロードがONTAPでサポートされるよりも多くの同時処理を送信する場合に、システム内の他のワークロードに与える影響を制限するのに役立ちます。この新機能は、1つのワークロードが他のワークロードを過負荷にしないように、すべての接続で使用可能なEXECコンテキストの数を抑制することで機能します。このスロットリングは、ノード上のEXECコンテキストの合計使用率に基づいており、ノードの合計が超過しないように操作をスケールバックするのに役立ちます。

表19）Execコンテキストのスロットルスケール

EXECコンテキストのノード使用率	スケールバックファクタ	接続ごとのEXEC制限
60%	1	128
70%	8	16
80%	16	8

ノードが使用可能なEXECコンテキスト全体の70%に達すると、各接続は、ノードの合計使用率が60%に戻るまで、16の同時操作しか実行できません。EXECの上限は128に戻ります。

いくつかの考慮事項：

- この機能はONTAP 9.9.1以降でのみ使用でき、256GBを超えるメモリを搭載したプラットフォーム（AFF A700、AFF A800、FAS9000など）でのみ使用できます。
- これらのプラットフォームでは、ノードあたりの使用可能なEXECコンテキストの総数も10,000に増加します。6、000を超える実行が割り当てられるまで調整は行われないため（以前のEXECの合計上限は3、000）、既存のワークロードでパフォーマンスに悪影響が及ぶことはありません。
- このスロットリングは、クライアントごとのオーバーランによってブロックされるEXECコンテキストの数を減らすのには役立ちません。Linuxクライアントの調整やnconnectの使用については、「Network connection concurrency and TCP slots：NFSv3」セクションのガイダンスに従う必要があります。

接続が調整されているかどうかを確認するにはどうすればよいですか。

メモリが256GB以上のプラットフォームでONTAP 9.9.1を実行している場合、EXECスロットリングがデフォルトで有効になります。それ以外のプラットフォームでは、この機能は適用されません。

クラスタでEXECスロットリングが発生しているかどうかを表示するには、 `exec_ctx` 統計オブジェクトを起動します。

```
cluster::> set diag
cluster::*> statistics start -object exec_ctx
```

次に、統計を表示します。EXECスロットリングが使用されている場合は、次のカウンタが表示されます。

- **throttle_scaleの略。**現在有効なスケールバックファクタ（1、8、または16）。
- **throttle_increases**スケールバックファクタが増加した回数。
- **スロットル減少**スケールバックファクターが減少した時間。
- **throttle_hist :** 各スケール係数での割り当てのヒストグラム（1、8、または16のカウンタの増分）。

ネットワーク接続の同時実行数とTCPスロット：NFSv3

クライアントからONTAP NFSサーバへのNFSマウントが確立されると、CIDも確立されます。受信する各NFS処理には、ONTAPで実行コンテキスト（`exec_ctx` または `exec`）と呼ばれるプレースホルダリソースが割り当てられます。処理が完了すると、リザーブされていたが `execs` システムに解放され、新しい受信処理で使用できるようになります。

ONTAPでは、CIDごとに `execs` 任意の時点で128を使用できます。クライアントが一度に128を超える処理を送信すると、ONTAPは新しいリソースが解放されるまでそのクライアントにプッシュバックします。ほとんどの場合、このようなプッシュバックは1処理あたりマイクロ秒にすぎませんが、数百ものクライアントから数百万の要求が送信されると、プロトコル、ディスク、ノードのレイテンシなど、ストレージシステムのパフォーマンス問題の兆候が通常見られないパフォーマンスの問題が発生する可能性があります。そのため、これらの問題の切り分けは困難で時間がかかる場合があります。

以前のLinuxカーネル（RHEL 6.xより前のリリース）では、RPCスロットテーブルの静的な設定は16でした。新しいLinuxクライアントでは、この設定が最大65、536に変更され、NFSクライアントによって必要なスロットテーブルの数が動的に増加しました。その結果、新しいNFSクライアントは、一度に処理できるよりも多くの要求でNFSサーバをフラッディングさせる可能性があります。

NFSv4.x処理は複合要求（パケットごとに3つまたは4つのNFS処理など）として送信され、NFSv4.xセッションスロットはRPCスロットテーブルの代わりに要求の並列化に使用されます。詳細については、「NFSv4.x concurrency−session slots」を参照してください。

スロットテーブルによって発生するパフォーマンス問題に対処するには、次の方法があります。

- クライアントごとにNFSv3マウントポイントを追加する（有効にするにはボリューム内の別の場所を使用する）
- 1つのクライアントがTCP接続/セッションごとに送信できるNFSv3要求の数を調整する
- マウントあたりのTCP接続/セッション数を増やすには、nconnectマウントオプションを使用します（クライアントとONTAPの両方のバージョンでnconnectがサポートされていることを確認してください）。

ただし、環境内のRPCスロットに対処する前に、NFSクライアントでRPCスロットテーブルを減らすことは実質的にはスロットルの一種であり、ワークロードによってはパフォーマンスに悪影響を及ぼす可能性があることに注意してください。100万個のNFS要求を送信する必要があるNFSクライアントは、RPCスロットテーブルの設定に関係なく、これらの要求を送信します。RPCスロットテーブルを設定することは、基本的にNFSクライアントに、一度に送信できる要求の数を制限するように指示します。NFSクライアントをスロットルするか、ストレージシステムでフロー制御を実行するかの決定は、ワークロードとユースケースによって異なります。

これらの値を調整する前に、スロットテーブルが多すぎると原因のパフォーマンスの問題やアプリケーションへの影響が発生するかどうかをテストして特定することが重要です。

RPCスロットテーブルの潜在的な問題の特定

前述したように、最新のNFSv3クライアントではRPCスロットテーブルに動的な値が使用されます。つまり、クライアントは1つのTCPセッションでできるだけ多くの同時処理（最大65、336）を送信します。ただし、ONTAPでは1つのTCP接続で同時に実行できる処理は128しかないため、クライアントからの送信数が128を超えると、ONTAPはNFSv3処理に対してフロー制御を実行し、リソースが解放されるまでNFS処理（ONTAPのEXECコンテキスト）をブロックして、不正なクライアントがストレージシステムを過負荷にするのを防ぎます。このフロー制御はパフォーマンスの問題として顕在化する可能性があり、原因の余分なレイテンシやジョブの完了時間の短縮など、一般的なストレージシステムの統計には明確な理由がない可能性があります。ネットワーク関連の問題のように見える場合があり、ストレージ管理者が誤ったトラブルシューティングパスをたどる可能性があります。

RPCスロットテーブルが関係しているかどうかを調べるには、ONTAPパフォーマンスカウンタを使用します。オーバーランされている接続によってブロックされたEXECコンテキストの数が増加しているかどうかを確認できます。

これらの統計を収集するには、次のコマンドを実行します。

```
statistics start -object cid -instance cid
```

次に、一定期間の統計を確認して、統計が増加しているかどうかを確認します。

```
statistics show -object cid -instance cid -counter execs_blocked_on_cid
```

NFSクライアントでは `nfsiostat`、コマンドを使用してアクティブな実行中スロットテーブルを表示できます。

```
# nfsiostat [/mount/path] [interval seconds]
```

クライアントでRPCスロットテーブルの値を小さく設定すると、RPCスロットテーブルのキューがストレージからクライアントに移動するため、`rpc bklog` 値は大きくなります。

スロットテーブルが128に設定されている場合、は `rpc bklog` 2つのクライアントから500万個のファイルを作成し、約26,000回/秒の処理を送信したときに360度の高さになりました。

```
# nfsiostat /mnt/FGNFS 1 | grep "bklog" -A 1
           ops/s       rpc bklog
       25319.000         354.091
--
           ops/s       rpc bklog
       24945.000         351.105
--
           ops/s       rpc bklog
       26022.000         360.763
```

しかし、ONTAPは受信操作をブロックする必要はありませんでした。

```
cluster::*> statistics show -object cid -counter execs_blocked_on_cid -sample-id
All_Multi1_bs65536

    Counter                                                     Value
    -------------------------------------------- ----------------------------------------
    execs_blocked_on_cid                                            0

    Counter                                                     Value
    -------------------------------------------- ----------------------------------------
    execs_blocked_on_cid                                            0
```

RPCスロットテーブルの値が高い値に設定されている場合、RPCキュー（`rpc bklog`）はクライアント上で低くなります。この場合、スロットテーブルはデフォルト値65,536のままです。クライアントのバックログが0であり、処理数が高くなっています。

```
# nfsiostat /mnt/FGNFS 1 | grep "bklog" -A 1
           ops/s       rpc bklog
       22308.303           0.000
--
```

```
          ops/s        rpc bklog
       30684.000            0.000
```

つまり、クライアントが保持している処理の数がそれほど多くないため、ストレージがこれらのRPC呼び出しをより多く吸収する必要があります。これはONTAPの統計で見ることができます。

```
cluster::*> statistics show -object cid -counter execs_blocked_on_cid -sample-id
All_Multi1_bs65536

    Counter                                                     Value
    -------------------------------------------- ----------------------------------------
    execs_blocked_on_cid                                       145324

    -------------------------------------------- ----------------------------------------
    Counter                                                     Value

    execs_blocked_on_cid                                       124982
```

ブロックされたexecの数が一定数を超えると、EMSがログに記録されます。次のEMSが生成されました。

```
cluster::*> event log show -node tme-a300-efs01-0* -message-name nblade.execsOverLimit
Time                Node              Severity      Event
------------------- ----------------- ------------- ---------------------------------------------------------------------
4/8/2021 17:01:30   node1
                                      ERROR         nblade.execsOverLimit: The number of in-flight
requests from client with source IP x.x.x.x to destination LIF x.x.x.x (Vserver 20) is greater
than the maximum number of in-flight requests allowed (128). The client might see degraded
performance due to request throttling.
```

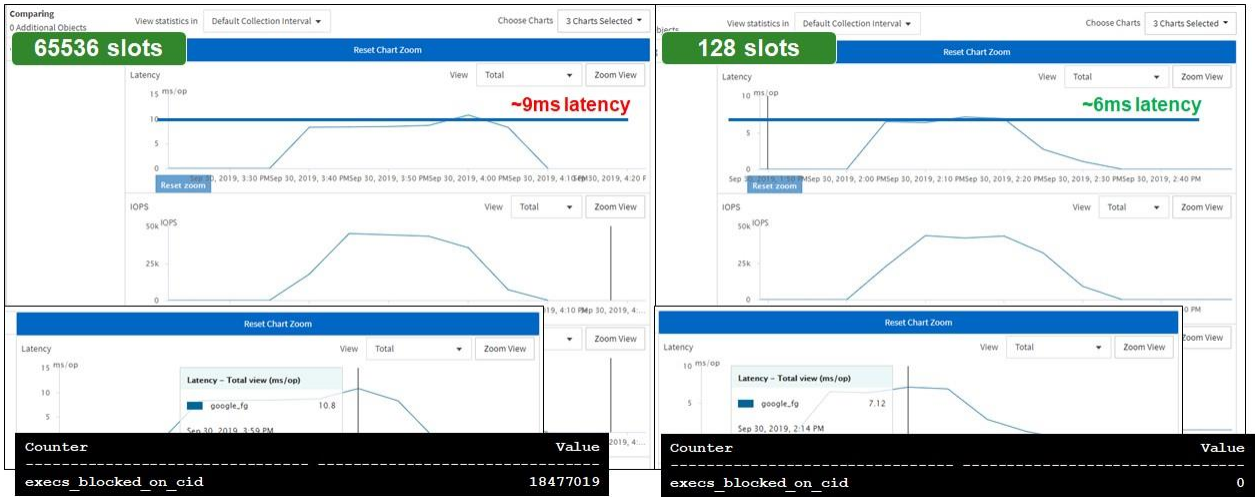
通常、 `nblade.execOverLimit` ONTAP 9.8以降でEMSイベントが表示されない場合は、RPCスロットテーブルが原因でワークロードに問題が発生している可能性はありません。ONTAP 9.7およびealrlierでは、これらのイベントは存在しないため、CIDの統計情報を監視し、の増加を監視する必要があります `exec_blocked_on_cid`。問題が使用されているかどうか不明な場合は、NetAppサポートにお問い合わせください。

例1：RPCスロットテーブルがパフォーマンスに与える影響–ファイル数の多いワークロード

図19の例では、18万個のサブディレクトリに1,800万個のファイルを作成するスクリプトを実行しました。この負荷生成は、3つのクライアントから同じNFSマウントに対して実行されました。その目的は、原因ONTAPに対するデフォルトのRPCスロットテーブル設定を持つクライアントで、フロー制御シナリオを開始するのに十分な数のNFS処理を生成することでした。その後、同じスクリプトが同じクライアントで再度実行されましたが、RPCスロットテーブルは128に設定されています。

その結果、デフォルトのスロットテーブル（65,536）では1、800万件 `execs_blocked_on_cid` のイベントが生成され、RPCスロットテーブルの設定（128）での実行と比較してワークロードに3ミリ秒のレイテンシが追加されました。

図19）RPCスロットテーブルがNFSv3のパフォーマンスに与える影響



3ミリ秒はそれほどのレイテンシとは思えないかもしれませんが、数百万回を超える処理を合計すると、ジョブの完了が大幅に遅くなる可能性があります。

例2：RPCスロットテーブルがパフォーマンスに与える影響–シーケンシャルI/Oワークロード

「さまざまなTCP最大転送ウィンドウサイズのパフォーマンス例」のセクションでは、NFSv3とNFSv4.1のパフォーマンスの違い、wsize / rsizeマウントオプションの値の違い。また、これらのテストの実行中に、RPCスロットテーブルによってCIDでブロックされるexecの数が増加すると、パフォーマンスのボトルネックが発生し、一部のパフォーマンス実行に14.4ミリ秒の書き込みレイテンシが追加され、ジョブ全体の完了時間が5.5分延長されたこともわかりました。

テストは、複数のdd処理を並行して実行するスクリプトを使用して、10GBネットワーク上の2つのクライアントで実行しました。全体的に、それぞれ2つの16GBファイルを含む8つのフォルダが作成され、読み取られて削除されました。

- RPCスロットが最大ダイナミック値65,536に設定されている場合、dd操作には20分53秒かかりました。
- RPCスロットを128に下げると、同じスクリプトに15分23秒しかかかりませんでした。

1MBのwsize/rsizeマウントオプションと65、536個のRPCスロットを使用した場合、はexecs_blocked_on_cidノードあたり約1、000に増分されます。

```
cluster::*> statistics show -object cid -counter execs_blocked_on_cid

Scope: node1
    ------------------------------------------- -------------------------------------------
     Counter                                                     Value

     execs_blocked_on_cid                                         1001

Scope: node2
    ------------------------------------------- -------------------------------------------
     Counter                                                     Value

     execs_blocked_on_cid                                         1063
```

図20 は、1MBのwsize/rsizeマウント値を使用したジョブのレイテンシ、IOPS、およびスループットの比較を示しています。

図20）並列ddパフォーマンス–NFSv3およびRPCスロットテーブル、1MB rsize/wsize

図21 は、256Kのwsize/rsizeマウント値を使用したジョブのレイテンシ、IOPS、およびスループットの比較を示しています。

図21）並列ddパフォーマンス–NFSv3およびRPCスロットテーブル、256K rsize/wsize

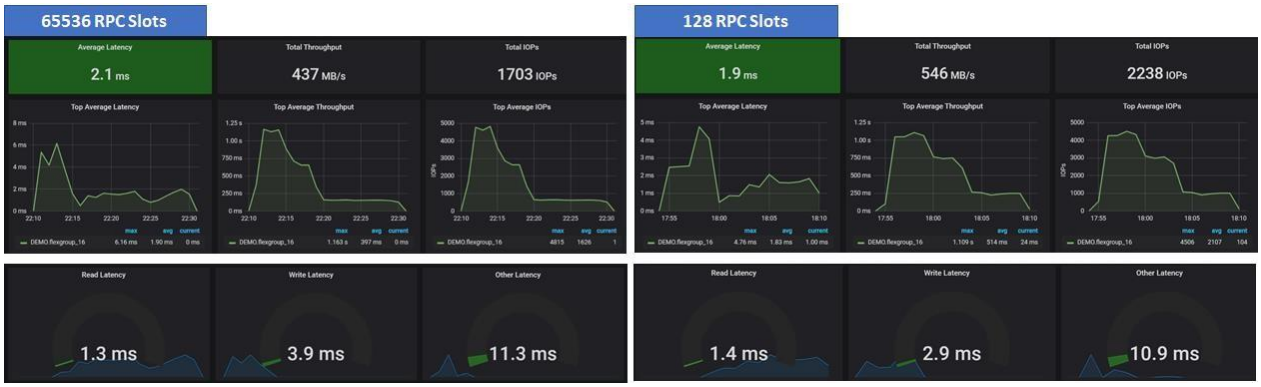


表20 に、65,536および128 RPCスロットでのジョブ時間とレイテンシの比較を示します。

表20）ジョブの比較–65、536、128 RPCスロットのパラレルdd

テスト	平均読み取りレイテンシ（ミリ秒）	平均書き込みレイテンシ（ミリ秒）	完了時間
NFSv3-1MB wsize/rsize、65、536スロットテーブル	9.2（+3.2ミリ秒）	42.3（+14.4ミリ秒）	20m53s（+5m30s）
NFSv3-1MB wsize/rsize、128スロットテーブル	6	27.9	15分23秒
NFSv3-256K wsize/rsize、65、536スロットテーブル	1.3（-.1ミリ秒）	3.9（+1ミリ秒）	19分12秒（+4分55秒）
NFSv3-256K wsize/rsize、128スロットテーブル	1.4	2.9	14分17秒
NFSv3-64K wsize/rsize、65、536スロットテーブル	0.2（0ミリ秒）	3.4（+1.2ミリ秒）	17m2s（+2m14s）
NFSv3-64K wsize/rsize、128スロットテーブル	2	2.2	14m48

RPCスロットテーブルに関する問題の解決

場合によっては、カウンタの増分によってパフォーマンス問題が作成されない可能性があるため、NFSクライアントのスロットテーブルをまったく調整する必要がないことがあります。クライアントが100万件の要求をONTAP NFSマウントに送信する場合、クライアントが送信するように調整されている内容に関係なく、クライアントはそれらの要求を送信することに注意してください。ここで考慮すべきことは、プッシュバックはどこから来るのか、つまりクライアントとサーバのどちらから来るのか、ということです。

RPCスロットテーブルを128に設定すると、クライアントは一度に128個の同時要求だけをストレージシステムに送信し、余分な要求はすべて独自のキューに保持します。`nfsiostat` コマンドで確認できます（の例について `nfsiostat`は、「RPCスロットテーブルの潜在的な問題の特定」を参照してください）。

場合によっては、この値によってONTAPの増分EXECカウンタが停止し、EMSメッセージが抑制されることがありますが、クライアントの設定や使用可能なリソースによっては、実際にパフォーマンスが低下することがあります。

ONTAPでは、NFSv3処理でクライアントがTCPセッションごとに送信するスロットテーブルの数を制御できません。したがって、クライアントスロットテーブルの設定が原因でパフォーマンス問題が発生した場合は、NFS経由で送信されるスロットテーブルの最大数を128以下に制限するようにクライアントを設定する必要があります。この設定の構成は、クライアントOSのバージョン、クライアントの数によって異なり、この設定を有効にするには再マウントが必要です。クライアントOSの値を設定する方法の詳細については、クライアントベンダーにお問い合わせください。

前述したように、RPCスロットテーブルをに設定する値は、ワークロードとクライアントの数によって異なります。たとえば、上記のテストでは、1MBのwsize/rsizeを使用した128個のRPCスロットが、ジョブ完了時間15分23秒を達成しました。RPCスロットを64に設定すると、ジョブの完了時間が14分2秒に短縮されました。ただし、この値を低く設定すると逆の効果があり、クライアントは不必要に調整されます。可能な限り最良の結果を得るために、環境内のさまざまな値をテストします。

表21）ジョブの比較–65、536、128、および64 RPCスロットのパラレルdd

テスト	平均読み取りレイテンシ（ミリ秒）	平均書き込みレイテンシ（ミリ秒）	完了時間
NFSv3-1MB wsize/rsize、65、536スロットテーブル	9.2（+3.2ミリ秒）	42.3（+19.4ミリ秒）	20m53s（+6m51s）
NFSv3-1MB wsize/rsize、128スロットテーブル	6（-.3ミリ秒）	27.9（+5ミリ秒）	15分23秒（+1分21秒）
NFSv3-1MB wsize/rsize、64スロットテーブル	6.3	22.9	14m2秒

同じクライアント上のクラスタ内の別々のIPアドレスに追加のマウントポイントを接続することで、クライアントのNFS接続のパフォーマンスを向上させることは可能ですが、この方法では複雑さが増す可能性があります。たとえば、にボリュームをマウントする代わりに、 `SVM:/volumename`同じクライアント上に複数のマウントポイントを作成して、異なるフォルダやボリューム内のIPアドレスに対応させることができます。

例：

```
LIF1:/volumename/folder1
LIF2:/volumename/folder2
LIF3:/volumename/folder3
```

nconnectを使用したスロットテーブルの枯渇の問題の回避

もう1つの方法として、 同じTCP接続でNFSv3の多重化を実行できる一部のLinuxディストリビューションでは、nconnectマウントオプションを使用できます。このオプションを使用すると、使用可能な同時セッション数が増え、全体的なパフォーマンスが向上します。たとえば、nConnect=8を使用すると、RPCスロット×8セッションが128になり、最大1、024スロットの同時実行が可能になります。これはまた、クライアント側のスロットテーブル設定を調整する必要性を減らすのに役立ちます。

たとえば、クライアントのデフォルトの65、536スロットテーブルを使用してNFSマウントにnconnectを適用しなかった場合のテストの結果は次のとおりです。

- 2つのクライアントが100万個の4Kファイルと2、000個のディレクトリを81秒以内に作成
- 同じテストでnConnect=2を使用した場合、2つのA300 AFFノードでブロックされた合計835、324個の実行結果は次のとおりです。

- 2つのクライアントが100万個の4Kファイルと2、000個のディレクトリを82秒以内に作成
- 2つのA300 AFFノードで合計827、275個のExecsをブロック

では、テストはnconnect=8で実行されました。

- 2つのクライアントで100万個の4Kファイルと2、000個のディレクトリを85秒以内に作成
- 2つのA300 AFFノードでブロックされるExecの総数はゼロ

使用するファイル作成スクリプトは、
https://github.com/whyistheinternetbroken/NetAppFlexGroup/blob/master/file-create-hfc.pyから入手できます。

表22 に、さまざまなnconnect設定とその結果を使用して、この特定のテストを並べて比較したものを示します。また、前述の注意事項についても説明します。`exec contexts` ONTAPでブロックされた場合、パフォーマンスの低下とは必ずしも同じではありません。また、nconnectは、指定したセッション数と使用中のワークロードに基づいてパフォーマンスを向上させることができます。この一連のテストはすべて、新しい EXECコンテキストスロットル機能を含むONTAP 9.9.1に対して実行されました。それぞれ10回のランが使用され、平均化され、デフォルトのクライアント設定が使用されました。使用したマウントwsizeとrsizeは64Kです。

ワークロードの構成は次のとおりです。

```
    create_percent                                  33%
    lookup_percent                                  33%
    write_percent                                   33%
```

表22）ファイル数の増加（100万ファイル）–NFSv3–nconnectあり/なし–デフォルトスロットテーブル

テスト	平均完了時間	ONTAPでブロックされた平均総実行数
NFSv3-nconnectなし	約69.5秒	214770
NFSv3-nconnect=2	約70.14秒	88038
NFSv3-nconnect=4	約70.1秒	11658
NFSv3-nconnect=8	約71.8秒	0
NFSv3-nconnect=16	約71.7秒	0

これらのテストでは、次のことを確認できます。

- この種のワークロードでは、nconnectはあまり役に立ちません。
- EXECコンテキストがブロックされると、パフォーマンスの問題が発生する可能性があります（「例1：RPCスロット テーブルによるパフォーマンスへの影響–ファイル数の多いワークロード」を参照）が、必ずしもパフォーマンスの問題が発生するとは限りません。

同じテストを、最大128のRPCスロットテーブル設定を使用して、nconnectなしでnconnect = 8の場合にのみ再実行しました。この例では、クライアントでRPCスロットテーブルを調整しても、このワークロードの平均完了時間のパフォーマンスにわずかなプラスの影響があるだけでなく、予測性も向上しました。たとえば、スロットテーブルが65,536に設定されている場合、10回の実行の完了時間は、このワークロードでは55.7秒から81.7秒という大きな違いがありましたが、128のスロットテーブルでは68~71秒の一貫したパフォーマンス範囲が維持されました。これは、ストレージがNFS処理をプッシュバックする必要がないためです。混在環境にクライアントを追加すると、1つまたは2つのクライアントが他のクライアントをいじめてパフォーマンスが低下する可能性があるため、予測の影響力が高まります。特に9.9.1より前のONTAPリリースでは、この傾向が顕著になります。（影響源のワークロードを軽減するためにEXECコンテキストスロットリングが追加されました）。

このテストにnconnectを追加したときは、クライアントが128に調整されていたため、各クライアントが複数のTCPセッションでNFS処理をプッシュできなかったため、パフォーマンスが少し低下しました。ワークロードにnconnectを使用する場合は、クライアントにRPCスロットテーブルの値を設定せずに、nconnectでTCPセッション全体にワークロードを分散させることを検討してください。

表23）ファイル数の増加（100万ファイル）–NFSv3–nconnectあり/なし–128スロットテーブル

テスト	平均完了時間	ONTAPでブロックされた平均総実行数
NFSv3-nconnectなし	約69.4秒	0
NFSv3-nconnect=8	約71.2秒	0

スロットテーブルをさらに（16に）縮小すると、クライアントが一度に16個の要求を送信するだけであるため、nconnectなしでスクリプトが完了するまでに約28.1秒以上かかるため、パフォーマンスが大幅に低下しました。nconnectでは、1つのセッションで16個のスロットテーブルではなく、セッションあたり16個のスロットテーブル（16×8 = 128）が得られるため、より多くのスロットテーブル（約70.6秒）を使用した他のテストとほぼ同じ平均完了時間を維持できました。この構成でnconnectを使用すると、セッションごとに128の処理を送信でき、セッションごとに16の処理しか送信できないため、この構成でnconnectを使用すると、パフォーマンスが大幅に向上します。ただし、数百のクライアントがある環境では、スロットテーブルのオーバーランによるパフォーマンスの問題を回避するには、スロットテーブルを16に設定するしかありません。

表24）ファイル数の増加（100万ファイル）–NFSv3–nconnectあり/なし–16スロットテーブル

テスト	平均完了時間	ONTAPでブロックされた平均総実行数
NFSv3-nconnectなし	約99.3秒	0
NFSv3-nconnect=8	約70.6秒	0

これらの結果から、nConnectは、パフォーマンスを大幅に犠牲にすることなく、高いパフォーマンスと多数のNFS操作を必要とする環境でRPCスロットテーブルを調整する必要がなくなることなく、TCP接続間でRPCスロット要求をより適切に分散する方法を提供できます。nconnectの詳細については、を参照してくださいnconnect。

注： ワークロードにはさまざまな種類があり、パフォーマンスへの影響があるため、単一の正しいNFS設定が存在しないため、環境ではさまざまなクライアント設定やマウントオプションなどを常にテストする必要があります。

クライアント数の多い環境でのRPCスロットテーブルの設定

TCP接続ごとのスロットテーブルのONTAPセッションの制限は128ですが、EXECコンテキストのノードレベルの制限を超えることもできます。

たとえば、1つのノードで最大3,000のEXECコンテキストを処理でき、各TCPセッションで最大128のEXECコンテキストを処理できる場合、これにより、リソースが使い果たされる前に任意の時点で最大24の同時TCPセッションがRPCスロットテーブルを最大にすることができ、ONTAPはリソースを解放して使用できるようにするために一時停止する必要があります（3、000/128 = 23.47）。ノードごとに使用可能なEXECコンテキストを確認する方法の例については、「Execコンテキストスロットリング」を参照してください。

これは、ONTAPがノードあたり最大24のクライアントしかサポートできないことを意味するわけではありません。ONTAPは、ノードあたり最大100,000のTCP接続をサポートします（プラットフォームによって異なります）。つまり、24台以上のクライアントが1つのノードに同時に1つの接続あたり最大許容スロットエントリ（128）を送信すると、ONTAPがリソースを解放するために遅延が発生します。

表25 に、接続あたりの最大同時処理数を1つのノードに送信できるクライアント数の例を示します。

表25）ノードEXECコンテキストが枯渇するまでの最大同時操作数（128）の合計クライアント数

ノード タイプ	ノードあたりの使用可能なEXECコンテキストの総数	ノードあたりの接続あたりの最大送信処理数（128）
AFF8040	1,500	1,500/128 =~11.7
AFF A300	3,000	3,000/128 =~23.4
AFF A800	10,000	10,000/128 =~78.1 *

注： *この数は、EXECコンテキストスロットリングによって異なります。

クライアントのRPCスロットテーブルの最大エントリ数が大きい場合、最大値に到達するために必要なクライアント数が少なくなります。環境内の数百のクライアントが、クラスタ内の同じノードに対して同時に動作している場合（EDAワークロードなど）、その場合は、クライアントの数とワークロードに参加しているクラスタノードの数に基づいて、RPCスロットテーブルの値をはるかに低く設定することを検討してください。

表26）ノードEXECコンテキストが枯渇する前に16スロットテーブルを使用した合計クライアント数

ノード タイプ	ノードあたりの使用可能なEXECコンテキストの総数	ノードあたりの接続あたりの最大送信処理数（128）
AFF8040	1,500	1,500/16 =~93.7
A300	3,000	3,000/16 =~187.5
A800	10,000	10,000/16 = 625 *

また、次のアプローチを使用すると、多数のクライアントのワークロードの全体的なパフォーマンスを向上させることができます。

- ノードとデータLIFの数が多いクラスタ。
- DNSラウンドロビン：初回マウント時にネットワーク接続をより多くのノードに分散します。
- FlexGroupボリュームを使用すると、クラスタ内のより多くのハードウェアを利用できます。
- 読み取り負荷の高いワークロードをより多くのノードとマウントポイントに分散しますFlexCache。
- 同時処理数を減らすために、NFSクライアントでRPCスロットテーブルの値を低く設定することを検討してください。値はプラットフォームやONTAPのバージョンによって異なり、クライアント数によって異なります。たとえば、「表25」を参照してください。」
- nconnectを使用すると、単一クライアントのパフォーマンスが向上します。
- ONTAP 9.9.1以降：256GB以上のRAMを搭載したプラットフォームを使用している場合は、「Execコンテキスト スロットル」を使用してワークロードへの影響の影響を軽減することができます。

スロットテーブルの推奨事項は、ONTAPハードウェア、ONTAPバージョン、NFSマウントオプションなどに基づいて調整されます。環境内で異なる値をテストすることを強く推奨します。RPCスロットテーブルの制限は他のNASプロトコルに影響しますか。

RPCスロットテーブルの制限はNFSv3トラフィックのみに影響します。

- SMBクライアントでは、SMBマルチチャネル、SMB多重化、SMBクレジットなど、さまざまな接続方法を使用して同時実行を実現します。SMBの接続方法は、クライアント/サーバの設定とプロトコルのバージョンによって異なります。たとえば、SMB 1.0ではSMB多重化（mpx）が使用され、SMB2.xではSMBクレジットが使用されます。
- NFSv4.xクライアントはRPCスロットテーブルを使用せず、状態IDとセッションスロットを使用してクライアントからの同時トラフィックのフローを制御します。

表27 に、同じ65536スロットテーブル値を使用したNFSv3とNFSv4.1のテスト実行結果を示します。

表27）ジョブの比較- parallel dd–NFSv3およびNFSv4.1、65536 RPCスロット

テスト	平均読み取りレイテンシ（ミリ秒）	平均書き込みレイテンシ（ミリ秒）	完了時間
NFSv3-1MB wsize/rsize、65、536スロットテーブル	9.2（+2.7ミリ秒）	42.3（+5.5ms）	20m53（+5m47）
NFSv4.1-1MB wsize/rsize、65、536スロットテーブル	6.5	36.8	15分6秒
NFSv3-256K wsize/rsize、65、536スロットテーブル	1.3（-.1ミリ秒）	3.9（+.7ミリ秒）	19分12秒（+7分2秒）
NFSv4.1-256K wsize/rsize、65、536スロットテーブル	1.4	3.2	12分10秒
NFSv3-64K wsize/rsize、65、536スロットテーブル	0.2（+.1ミリ秒）	3.4（+2.2ミリ秒）	17 m2秒（+1 m54秒）

テスト	平均読み取りレイテンシ（ミリ秒）	平均書き込みレイテンシ（ミリ秒）	完了時間
NFSv4.1-64K wsize/rsize、65、536スロットテーブル	.1	1.2	15 m8秒

仮想メモリの調整

NFSクライアントを調整してパフォーマンスを向上させるもう1つの方法は、仮想メモリの設定とダーティバッファの値を変更することです（vm.dirty）。仮想メモリキャッシュは、クライアントがファイルキャッシュを実行する方法に重要です。また、デフォルト値を変更することで、クライアントでの読み取りまたは書き込みのパフォーマンスを制御できます。クライアントは、ファイル操作をディスクにフラッシュする前にRAMに直接書き込みます。そのため、NFSクライアントとサーバの間でワークロードに関するやり取りが少なくなります。詳細については、https://www.kernel.org/doc/Documentation/sysctl/vm.txtを参照してください。

ファイルキャッシュのフラッシュには、次の3つのトリガーがあります。

- **時間ベース：**バッファがこの調整可能で定義された経過時間に達したら、クリーニング用にマークを付ける必要があります（フラッシュなど、ストレージへの書き込みとも呼ばれます）。
- **メモリ圧力。**割り当てられたメモリがいっぱいになると、ファイルキャッシュがフラッシュされます。
- **閉めて**ファイルハンドルを閉じると、すべてのダーティバッファが非同期でストレージにフラッシュされます。

クライアントのデフォルトのキャッシュ設定では、通常、ほとんどのワークロードに適した、ファイルキャッシュ専用のRAMの割合が提供されます。CentOS / RHEL 8.3クライアントのデフォルト値は次のとおりです。

```
vm.dirty_background_bytes = 0 ## modifying this sets vm.dirty_background_ratio to 0
vm.dirty_background_ratio = 10 ## modifying this sets vm.dirty_background_bytes to 0
vm.dirty_bytes = 0 ## modifying this sets vm.dirty_ratio to 0
vm.dirty_ratio = 30 ## modifying this sets vm.dirty_bytes to 0
vm.dirty_expire_centisecs = 3000
vm.dirty_writeback_centisecs = 500
vm.dirtytime_expire_seconds = 43200
```

ここでは、このセクションで説明する値の属性について説明します。これらの値は、クライアントを再マウントしたりリブートしたりしなくても設定できます。

vm.dirty_ratio|vm.dirty_bytes

これらの調整可能なデータは、変更されたものの、安定したストレージにまだ書き込まれていないRAMの容量を定義します。いずれかの調整可能に設定されている場合、は自動的にもう一方の調整可能なゼロを設定します。を vm.dirty_bytes 静的な値に設定すると、RAMの容量に関係なくクライアント全体でパフォーマンスの一貫性が向上しますが、ワークロードに利用可能なRAMが大量にあるシステムでは、パフォーマンスが人為的に制限される可能性もあります。

vm.dirty_background_ratio | vm.dirty_background_bytes

これらの調整可能なデータは、Linuxのライトバックメカニズムがダーティブロックを安定したストレージにフラッシュする開始点を定義します。

vm.dirty_expire_centisecs

この調整可能では、ダーティバッファが非同期的に書き出すためにタグ付けされるまでの時間を定義します。一部のワークロードでは、データの書き込み直後にファイルハンドルが閉じられないことがあります。ファイルを閉じないと、メモリ圧力または30秒が経過するまでフラッシュは発生しません（デフォルト）。これらの値を待機すると、アプリケーションのパフォーマンスに最適ではないことが判明するため、待機時間を短縮することで、一部のユースケースでパフォーマンスを向上させることができます。

vm.dirty_writeback_centisecs

カーネルフラッシャースレッドは、各フラッシュスレッドスリープ間でダーティバッファを非同期的にフラッシュします。この調整可能では、バッファフラッシュ間のスリープ時間を定義します。と組み合わせてこの値を小さくする `vm.dirty_expire_centisecs` と、一部のワークロードのパフォーマンスも向上します。

未調整のファイルシステムキャッシュのその他の影響

最新のシステムのデフォルトの仮想メモリ調整可能では、使用可能なRAMの量が考慮されない可能性があるため、ライトバックは、使用可能な仮想メモリを使用するように調整されていないファイルシステムキャッシュを含む他のストレージ処理の速度を低下させる可能性があります。

次のような潜在的な影響があります。

- ディレクトリのリスト表示が遅い、またはハングしている（ls）
- 読み取りスループットは書き込みスループットよりも大幅に低い
- nfsiostatからの高レイテンシ（秒以上）

これらの問題は、読み取りと書き込みの混在ワークロードを実行しているクライアントからのみ発生し、影響期間中は、そのクライアントのストレージエンドポイントからマウントされたすべてのNFSボリュームに発生します。

仮想メモリチューニングのメリットが最も大きいワークロードは何ですか。

ワークロードの種類は大きく変動する可能性があり、使用中のワークロードの種類とファイルの書き込み方法によって、仮想メモリの調整に与える影響が決まります。

表28では、Pythonスクリプトとf.write関数を使用して100万個のファイルが作成されました。テスト間の仮想メモリ設定には、次の変更が加えられました。

- `vm.dirty_ratio` 30%から40%に変更
- `vm.dirty_background_ratio` 10~20%
- `vm.dirty_expire_centisecs` 3,000~300
- `vm.dirty_writeback_centisecs` 500~100

このタイプの操作では、キャッシュが期限切れになるよりもはるかに速くファイルが閉じられるため、クライアントがキャッシュできることがほとんどないため、結果は期待を裏切るものでした。ただし、クライアントがより多くのファイルキャッシュをメモリに保持するように設定されている場合は、クライアントがストレージに多くの要求を頻繁に送信しないため、ONTAPでブロックされる実行回数が少なくなります。

表28）F.writeを使用した100万ファイル–NFSv3、65536スロット–VMダーティーバイトのデフォルトと調整済みの比較

テスト	平均完了時間	ONTAPでブロックされた平均総実行数	時間差
NFSv3-nconnectなし-VM.dirtyのデフォルト設定	約69.1秒	二一四、七七〇	-
NFSv3-nconnectなし-vm.dirty設定が調整されました	約69.5秒	一四四、七九〇	+0.4秒
NFSv3-nconnect=8-デフォルトのVM.dirty設定	約71.8秒	0	-
NFSv3-nconnect=8-vm.dirty設定を調整	約69.5秒	0	-2.3秒

ファイルサイズを大きくすると、仮想メモリの設定で全体の完了時間が大幅に改善されましたが、ここでのボトルネックはTCPセッションの制限によるものではないため、nconnectはそれほど大きな違いを生じませんでした。

このテストでは、ddを使用して50 x 500MB（2クライアント、フォルダあたり1ファイル、25フォルダ、クライアントあたり25プロセス）のファイルを作成し、wsize/rsizeマウント値を256Kに設定しました。

表29）ddを使用した500MBのファイル50個–NFSv3、65536スロット–VMダーティーバイトのデフォルトと調整済み

テスト	平均完了時間	ONTAPでブロックされた平均総実行数	時間差（vm.dirtyのデフォルトとVM.dirtyセット）
NFSv3-nconnectなし-VM.dirtyのデフォルト設定	約134.4秒	0	-
NFSv3-nconnectなし-vm.dirty設定が調整されました	約112.3秒	0	-22.1秒
NFSv3-nconnect=8-デフォルトのVM.dirty設定	約132.8秒	0	-
NFSv3-nconnect=8-vm.dirty設定を調整	約112.9秒	0	-19.9秒

ファイル数の多いファイルや小さなファイルの例にあるように、これらの設定はワークロードごとに大きな違いを生じません。正しい組み合わせが見つかるまで、さまざまな値をテストすることが重要です。

適切な値が見つかったら、を使用して `/etc/sysctl.conf` リブート時に値を保持できます。

NFSv4.x同時実行-セッションスロット

NFSv3ではTCPスロットテーブルでパフォーマンスが制限される可能性がありますが、NFSv4.xで接続の同時処理には独自の制限があり、NFSクライアントでのTCPスロットテーブルの設定はNFSv4.xの処理には適用されません。代わりに、NFSv4.xでは同時実行のためにセッションスロットが使用されます。

NFSv4.xセッションスロットも同様に動作し、1つのTCP接続を介してクライアントから送信される要求の数は、次のAdvanced Privilegeオプションで指定された設定値に制限されます。

```
[-v4.x-session-num-slots <integer>] - Number of Slots in the NFSv4.x Session slot tables
(privilege: advanced)
This optional parameter specifies the number of entries in the NFSv4.x session slot table. By
default, the number of slots is 180. The maximum value is 2000.
```

注： 記載されている最大数は2、000ですが、バグ1392470のようにNFSv4.xセッションがハングする可能性があるため、1、024を超えることは推奨されません。

NFSv4.xセッションが設定されると、クライアントとサーバは、そのセッションで許可される最大要求数をネゴシエートし、低い値（クライアントとサーバの設定）が適用されます。ほとんどのLinuxクライアントでは、デフォルトで64のセッションスロットが使用されます。この値は、modprobe設定を使用してLinuxクライアントで調整できます。

```
$ echo options nfs max_session_slots=180 > /etc/modprobe.d/nfsclient.conf
$ reboot
```

NFSv4.xマウントの現在の値は `systool` 、コマンド（sysfsutilsパッケージに含まれています）で確認できます。

```
# systool -v -m nfs | grep session
    max_session_slots   = "64"
```

クライアントオプションの変更後：

```
# systool -v -m nfs | grep session
    max_session_slots   = "180"
```

NFSサーバのデフォルト値は180ですが、ONTAPでは、接続ごとにCIDあたりのEXECコンテキスト数が128に制限されています。 そのため、NFSv4.xクライアントがONTAPへの単一のTCP接続で使用可能なEXECコンテキストをオーバーランし、値が高すぎるとリソースがシステムに解放される間に一時停止状態になる可能性があります。セッションスロットがオーバーランしている場合は `execs_blocked_by_cid`、

「RPCスロットテーブルの潜在的な問題の特定」セクションに記載されているカウンタも増加します。

ほとんどの場合、デフォルトの180値ではこの問題は発生しません。セッションスロットの値を高く設定すると、システムリソースが不足する可能性があります。クライアントのパフォーマンスを向上させるには、セッションスロット値を調整するのではなく、nconnectオプションを使用してこれらの操作をより並列に処理することを検討してください。

注： 次のような問題を回避するために、必ず最新のパッチが適用されたONTAP for NFSv4.xを使用してください。共通のクライアントTCPソケットからノードLIFへの複数の接続でNFSレイテンシが高い

セッションスロットを増やすと、全体的なパフォーマンスが向上しますか。

つまり、クライアントとサーバのセッションスロットを増やすと、ワークロードの同時処理性が向上するため、全体的なパフォーマンスが向上します。ただし、パフォーマンスが向上するかどうかは、ワークロードの種類、設定されたスロット数、クライアントの堅牢性、およびストレージシステムのビジー率によって異なります。また、1つのワークロードで利用可能なリソースが不足する可能性があるため、クライアントの他のワークロードにも影響します。異なるセッションスロット値をテストすることは、変更が環境に悪影響を与えないようにするために重要です。

表30では、ファイル数の多いワークロード（100万x 4KBファイル）と数の少ないシーケンシャルライトワークロード（2GBファイル×32）の完了時間、平均IOPS、スループットを測定しました。

使用されたパラメータは次のとおりです。

- 2つのAFF A300ノードにまたがるFlexGroupボリューム
- CentOS 8.3クライアント×2
- NFSv4.1マウント（AUTH_SYS、256K wsize、pNFSなし、nConnect=8）
- 仮想メモリのチューニングによる仮想メモリのチューニング

表30）NFSv4.xセッションスロットのパフォーマンスの比較

テスト	完了時間（秒）	平均 IOPS	平均MBps
NFSv4.1-100万x 4KBファイル（180セッションスロット）	～253.9	～11688	～15.2
NFSv4.1-100万x 4KBファイル（256セッションスロット）	～240.6	～12685	～16.7
NFSv4.1-100万x 4KBファイル（512セッションスロット）	～246.5	～12342	～16.1
NFSv4.1-100万x 4KBファイル（1、024セッションスロット）	～245.5	～12378	～16.3
NFSv4.1-2GBファイル×32（セッションスロット×180）	～148.3	~902	～224.5
NFSv4.1-2GBファイル×32（セッションスロット×256）	～149.6	889年まで	～221.5
NFSv4.1-32個の2GBファイル（512セッションスロット）	～148.5	891年まで	～222
NFSv4.1-2GBファイル×32（1、024セッションスロット）	～148.8	～898	～223.7

表31）NFSv4.xセッションスロットのパフォーマンス- 180スロットに対する変化率

テスト	完了時間	IOPS	MBps
NFSv4.1-100万x 4KBファイル（256セッションスロット）	-5.2%	+8.5%	+9.8%
NFSv4.1-100万x 4KBファイル（512セッションスロット）	-2.9%	+5.6%	+5.9%
NFSv4.1-100万x 4KBファイル（1、024セッションスロット）	-3.3%	+5.9%	+7.2%
NFSv4.1-2GBファイル×32（セッションスロット×256）	+0.9%	-1.4%	-1.3%
NFSv4.1-32個の2GBファイル（512セッションスロット）	+0.1%	-1.2%	-1.1%
NFSv4.1-2GBファイル×32（1、024セッションスロット）	+0.3%	-0.4%	-0.4%

所見

メタデータが多い/ファイル数が多い作成テストでは、セッションスロットの数が多いほど、完了時間、IOPS、スループットは全体的に向上しましたが、1、024のセッションスロットに対して256のセッションスロットが最適な状態にあるように見えました。

シーケンシャルライトワークロードでは、より高いセッションスロットを使用するとパフォーマンスがわずかに低下しました。これは、セッションスロットを増やすことが効果的であることを示していますが、すべてのワークロードタイプではありません。

NFSv4.xクライアントID / NFS4ERR_CLID_INUSE

NFSv4.xクライアントがONTAPでNFSエクスポートをマウントする場合、各クライアントはNFSマウントパケット内のRPCパケットの一部として自身のホスト名を送信します。NFSサーバは、NFSv4.xセッションを追跡するためにクライアントIDを返します。

ホスト名が同じ（IPアドレスが異なる）別のNFSv4.xクライアントがある場合、NFSサーバは次のエラーを返します。

```
50 11.856227 10.x.x.x 10.x.x.y NFS V4 Call (Reply In 51) EXCHANGE_ID
51 11.856407 10.x.x.y 10.x.x.x NFS V4 Reply (Call In 50) EXCHANGE_ID Status: NFS4ERR_CLID_INUSE
```

NFSv4.1 RFC標準では、これがセキュリティメカニズムである https://tools.ietf.org/html/rfc5661#section-2.10.8.3 です。

```
2.10.8.3. Protection from Unauthorized State Changes

As described to this point in the specification, the state model of NFSv4.1 is vulnerable to an
attacker that sends a SEQUENCE operation with a forged session ID and with a slot ID that it
expects the legitimate client to use next. When the legitimate client uses the slot ID with the
same sequence number, the server returns the attacker's result from the reply cache, which
disrupts the legitimate client and thus denies service to it. Similarly, an attacker could send
a CREATE_SESSION with a forged client ID to create a new session associated with the client
ID. The attacker could send requests using the new session that change locking state, such as
LOCKU operations to release locks the legitimate client has acquired. Setting a security policy
on the file that requires RPCSEC_GSS credentials when manipulating the file's state is one
potential work around, but has the disadvantage of preventing a legitimate client from releasing
state when RPCSEC_GSS is required to do so, but a GSS context cannot be obtained (possibly
because the user has logged off the client).

...

The SP4_MACH_CRED state protection option uses a machine credential where the principal that
creates the client ID MUST also be the principal that performs client ID and session maintenance
operations. The security of the machine credential state protection approach depends entirely on
safe guarding the per-machine credential. Assuming a proper safeguard using the per-machine
credential for operations like CREATE_SESSION, BIND_CONN_TO_SESSION, DESTROY_SESSION, and
DESTROY_CLIENTID will prevent an attacker from associating a rogue connection with a session, or
associating a rogue session with a client ID.
```

RFC-5661では、このエラーの意味は https://tools.ietf.org/html/rfc5661#section-15.1.13.2です。

```
 15.1.13.2. NFS4ERR_CLID_INUSE (Error Code 10017)
While processing an EXCHANGE_ID operation, the server was presented with a co_ownerid field that
matches an existing client with valid leased state, but the principal sending the EXCHANGE_ID
operation differs from the principal that established the existing client. This indicates a
collision (most likely due to chance) between clients. The client should recover by changing the
co_ownerid and re-sending EXCHANGE_ID (but not with the same slot ID and sequence ID; one or both
MUST be different on the re-send).
```

この場合、NFSv4.xクライアントは新しいクライアントIDを使用してマウントを再試行する必要があります。場合によっては、NFSマウントオプションの使用 `-clientaddr` が必要になることがあります。詳細については、RedHat Bugzilla #1821903を参照してください。

回避策

クライアントはRFC標準に従って期待どおりに動作しますが、この問題を回避する方法はいくつかあります。

回避策#1：同じSVMに異なるデータLIFがある

NFSv4.xクライアントID -問題は、同じホスト名のクライアントをSVMの同じデータLIFにマウントしようとしたときに発生します。ただし、追加のデータLIFを作成してマウントで使用できる場合、この問題は実行されません。Azure NetApp FilesやNetApp Cloud Volumes Serviceなどのマネージドクラウドサービスを使用する場合など、追加のデータLIFを作成できないことがあります。

回避策#2：異なるNFSv4.xバージョン

NFSv4.xクライアントID問題を回避するもう1つの方法は、あるクライアントが同じホスト名を持つ他のクライアントとバージョンが異なるボリュームをマウントする場合です。たとえば、client1（client.domain.com）がNFSv4.1を使用してマウントし、client2（client.domain.com）がNFSv4.2を使用してマウントした場合、この問題は発生しません。ただし、使用しているONTAPのバージョンが、試行するNFSのバージョンをサポートしている必要があります。ONTAP 9.8以降ではNFSv4.2がサポートされますが、Azure NetApp FilesやNetApp Cloud Volumes Serviceなどのマネージドクラウドサービスでは現在NFSv4.2がサポートされていません。

回避策#3：NFSv4.xマウントで使用するクライアント名を変更する

デフォルトでは、NFSv4.xはNFSサーバにマウントするときに、クライアントID値にクライアントのホスト名を使用します。ただし、デフォルトの動作を変更し、NFSv4.xマウントに使用されるクライアントIDを上書きするために使用できるクライアント側のNFSオプションがあります。

これを行うには、クライアントのNFSオプションnfs4-unique-idを、同じホスト名を使用するすべてのクライアントに対して静的な値に設定します（RedHat Bugzilla 1582186を参照）。この値を `/etc/modprobe.d/nfsclient.conf` ファイルに追加した場合、ファイルはリブート後も保持されます。クライアントの設定は次のように表示されます。

```
 # systool -v -m nfs | grep -i nfs4_unique
    nfs4_unique_id      = ""
```

設定するには、次のコマンドを実行します。

```
echo options nfs nfs4_unique_id=[string] > /etc/modprobe.d/nfsclient.conf
reboot
```

例：

```
# echo options nfs nfs4_unique_id=uniquenfs4-1 > /etc/modprobe.d/nfsclient.conf

# systool -v -m nfs | grep -i nfs4_unique
    nfs4_unique_id      = "uniquenfs4-1"
```

このオプションの詳細については、「NFSクライアント」を参照してください。

これはクライアント側の設定であるため、任意のONTAPバージョンで使用できます。これには、Azure NetApp FilesやNetApp Cloud Volumes Serviceなどのマネージドクラウドサービスが含まれます。

NFSノソツノナマエヘンコウ

NFSでは、ファイルがクライアントによってロックされているときにファイルが削除されると、NFSは「silly rename」と呼ばれる処理を実行し、ファイルの名前が `.nfs*` 名前に変更されます。これらのシナリオでは、アプリケーションがディレクトリを再帰的に削除する必要がある場合、ディレクトリは技術的に空ではないため、これらの削除は失敗します。

NFSv4.1 RFCでは、`OPEN4_RESULT_PRESERVE_UNLINKED` 単純な名前変更に対処するという新しい結果フラグが用意されています。

```
The use of the OPEN4_RESULT_PRESERVE_UNLINKED result flag allows a client to avoid the common
implementation practice of renaming an open file to ".nfs<unique value>" after it removes the
file. After the server returns OPEN4_RESULT_PRESERVE_UNLINKED, if a client sends a REMOVE
operation that would reduce the file's link count to zero, the server SHOULD report a value of
zero for the numlinks attribute on the file.
```

このフラグは現在、ONTAPのNFSv4.1実装ではサポートされていません。

ONTAP NFSによるatime更新の処理方法

アクセス時間（atime）は、NFSクライアント経由でファイルに最後にアクセスされた時刻です。これは `stat` 、Linuxのコマンドを使用して確認できます。次の例では `vmware-1.log` 、このクライアントのNFSマウント上のファイルのstatにアクセスタイムスタンプが2021年1月9日と表示されています。

```
# stat vmware-1.log
  File: ‘vmware-1.log’
  Size: 281796          Blocks: 560        IO Block: 65536 regular file
Device: 27h/39d Inode: 791804619   Links: 1
Access: (0600/-rw-------) Uid: (     0/    root)   Gid: (    0/    root)
Access: 2021-01-09 15:39:30.202594000 -0500
Modify: 2021-01-09 15:39:30.232596000 -0500
Change: 2021-01-14 15:40:29.946918292 -0500
```

などのコマンドを実行 `cat` すると、そのファイルのatimeが更新されます。

atimeは2021年1月14日に表示されます。

```
# cat vmware-1.log
# stat vmware-1.log
  File: ‘vmware-1.log’
  Size: 281796          Blocks: 552        IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 442679      Links: 1
Access: (0600/-rw-------) Uid: (     0/    root)   Gid: (    0/    root)
Access: 2021-01-14 15:41:11.603404196 -0500
Modify: 2021-01-09 15:39:30.232596000 -0500
Change: 2021-01-14 15:40:29.946918292 -0500
```

Linuxクライアントでは、クライアント側キャッシュが原因でatimeが正しく更新されない場合があります。などのコマンドでatimesが適切に更新されない場合は `cat`、NFSマウントを再マウントするか、クライアントのキャッシュを削除してみます（を実行すると `echo 1 > /proc/sys/vm/drop_caches`、クライアントのすべてのキャッシュが破棄されます）。CentOS 7.8のマニュアルページから：

```
The Linux client handles atime updates more loosely, however. NFS clients maintain good
performance by caching data, but that means that application reads, which normally update atime,
are not reflected to the server where a file's atime is actually maintained.
```

ONTAPは、クライアントからストレージシステムに更新が通知されるたびに、その都度更新されます。クライアントがONTAPに通知しない場合、atimeを更新する方法はありません。atime更新は、advanced権限のボリュームレベルオプションを使用して無効にできます `-atime-update`。

```
[-atime-update {true|false}] - Access Time Update Enabled (privilege: advanced)
This optionally specifies whether the access time on inodes is updated when a file is read. The
default setting is true.
```

NASフロー制御

ONTAPは、NASレベルのフロー制御も提供します。このフロー制御メカニズムは、データネットワークのNICおよびスイッチで有効になっているTCPフロー制御とは別のものです。常に有効な状態でNASプロトコル スタックに実装され、異常クライアントがクラスタ内のノードを過負荷にしてDoS攻撃を生成することを防止します。このフロー制御は、すべてのNASトラフィック（CIFSおよびNFS）に影響します。

仕組み

クライアントがノードに送信するパケットが多すぎる場合、フロー制御はウィンドウサイズをゼロに調整し、他のパケットが処理されるまで新しいNASパケットの送信を待機するようにクライアントに指示します。クライアントがこのゼロウィンドウの間もパケットの送信を継続する場合、NASプロトコルスタックフロー制御メカニズムはそのクライアントにTCPリセットを送信します。

NASフロー制御に関連するパフォーマンスの問題が疑われる場合は、NetAppテクニカルサポートに連絡してください。

FlexCloneを使用したボリューム内のすべてのUID / GIDの迅速な変更

ソフトウェア開発環境など、ファイル数の多いワークロードでは、一連のファイルが特定のユーザまたはグループに所有され、アクセスが制限されることがあります。別の開発者がこれらのファイルにアクセスする必要がある場合は、データセットの所有者/権限を変更する必要があります。数百万個のファイルとフォルダがある場合、その処理には長い時間がかかります。さらに、本番環境の他のユーザのアクセスを中断したくないため、この問題を中断することなく迅速に解決する方法が必要です。

NetApp FlexCloneボリュームを使用すると、ONTAPでスペースを消費することなく、Snapshotによってバックアップされたボリュームのコピーを瞬時に作成できます。また、クローン内のすべてのファイルおよびフォルダの所有権をすばやく変更することもできます。これらの変更を永続的に有効にするには、クローンを専用のボリュームにスプリットします（これにより、ONTAPの容量が使用されます）。次のコマンドを実行して、100万を超えるファイルとフォルダを含むボリュームのクローンを作成し、それらすべてのUIDとGIDを変更したところ、完了までに約10秒かかりました。

```
cluster::*> vol clone create -vserver DEMO -flexclone clone -parent-vserver DEMO -parent-volume
flexvol -junction-path /clone -uid 1100 -gid 1101
[Job 12606] Job succeeded: Successful
```

補助GID - NFSの16 GID制限への対処

NFSでは、1回のNFS要求で処理できる補助GID（セカンダリグループ）の最大数に特定の制限があります。AUTH_SYS/AUTH_UNIXの最大数は16、AUTH_GSS（Kerberos）の最大数は32です。これはNFSのプロトコルに関する既知の制限事項です。

ONTAPでは、要求元のユーザのグループをネームサービスからプリフェッチすることで、NFSパケットのグループリストが切り捨てられないようにすることで、補助グループの最大数を1、024に増やすことができます。

```
auth-sys-extended-groups
extended-groups-limit
```

仕組み

グループ数の制限を拡張するオプションは、他のNFSサーバのmanage-gidsオプションと同じように機能します。簡単に言うと、このオプションは、ユーザが属している補助GIDの全リストをダンプする代わりに、ファイルまたはフォルダでGIDを検索してその値を返します。

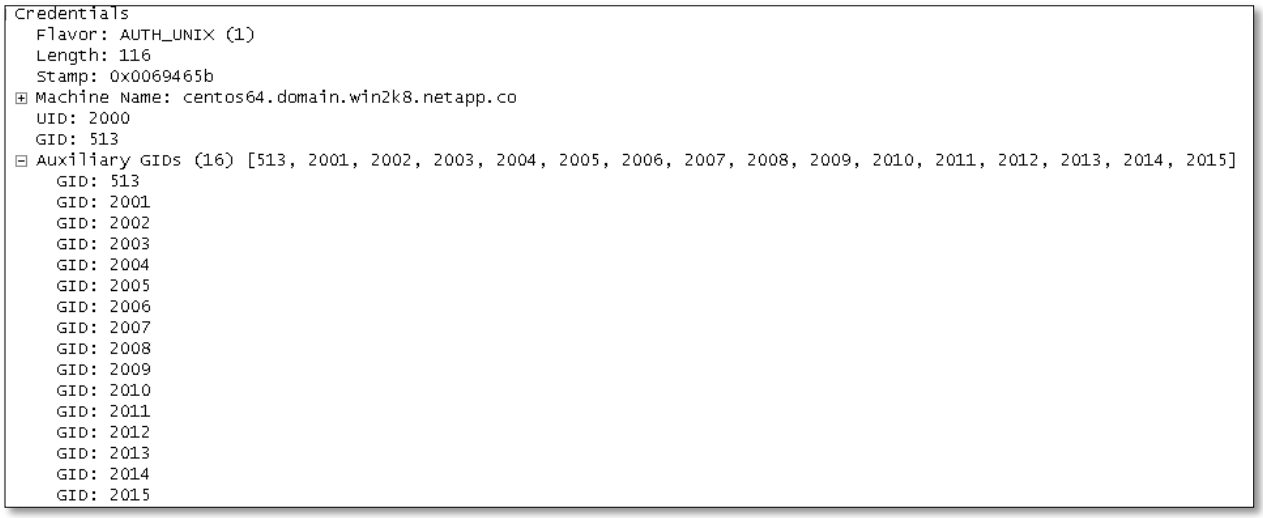
mountdのマニュアルページから、次の手順を実行します。

```
-g or --manage-gids

Accept requests from the kernel to map user id  numbers  into lists  of group  id  numbers for
use in access control. An NFS request will normally except when using Kerberos or other
cryptographic authentication) contains  a  user-id  and  a list of group-ids.  Due to a
limitation in the NFS protocol, at most 16 groups ids can be listed.  If you use the -g flag,
then the list of group ids received from the client will be replaced by a list of group ids
determined by an appropriate lookup on the server.
```

アクセス要求が行われると、パケットのRPC部分で渡されるGIDは16個だけです（図22）。

図22）16個のGIDを含むRPCパケット



制限値16を超えるGIDは、プロトコルによってドロップされます。拡張GIDは、外部ネームサービスで使用することも、ユーザとグループが適切に設定されている場合はクラスタでローカルに使用することもできます。ローカルUNIXユーザが複数のグループのメンバーであることを確認するには `unix-group adduser(s)`、次のコマンドを使用します。

```
COMMANDS
     adduser - Add a user to a local UNIX group

     addusers - Add a list of users to a local UNIX group
```

拡張GIDのパフォーマンスへの影響

グループを拡張してもパフォーマンスの低下は最小限に抑えられます（一般には1桁台前半のパーセンテージ）。メタデータNFSワークロードが大きいほど、特にシステムのキャッシュに対する影響が大きくなる可能性があります。パフォーマンスは、ネームサービスサーバの速度とワークロードによっても影響を受けることがあります。ネームサービスサーバが過負荷になると応答が遅くなり、GIDのプリフェッチに遅延が発生します。

ネームサービスの詳細については、TR-4668：『Name Service Best Practices』を参照してください。

数値ID認証に関する考慮事項（NFSv3およびNFSv4.x）

NFSv3でAUTH_SYSを使用すると、権限を解決するためにNFSマウントへのユーザ認証を実行するために、ユーザとグループに数値ID情報が送信されます。

ONTAPを搭載したNFSv4.xのマウントで名前文字列の代わりに数値ID文字列を使用できる機能があります。これにより、一元化されたネームサービス、クライアント/サーバ上の名前/数値IDの照合、IDドメインの照合などを必要とせずにNFSv4.xの処理を実行できます。(`-v4-numeric-ids`)。

この `-auth-sys-extended-groups` オプションを有効にすると、UNIXユーザの数値IDをネームサービスで有効なUNIXユーザ名に変換できない場合、数値ID認証が失敗します。これは、`- v4-numeric-ids` ONTAPが入力された数値ユーザIDを照会して、認証用の補助グループを検索する必要があるため、このオプションに対抗します。受信した数値IDを有効なUNIXユーザに解決できない場合、またはクライアントのUNIX数値UIDがONTAPが認識する数値と異なる場合は、 その後、`secd.authsys.lookup.failed` イベントログで検索が失敗し、ONTAPがクライアントにAUTH_ERRORを返し `client must begin a new session`ます。AUTH_ERRORは「Permission denied」と表示されます。

両方のオプションを使用するには、次のガイダンスに従ってください。

- NFSクライアントとサーバの両方から照会できないユーザとグループ、または数値IDが一致しないユーザとグループが必要な場合は、クライアントから数値IDではなく名前文字列が送信されるため、NFS KerberosとNFSv4.x ACLを利用してNFSv4.xで適切な認証を行うことができます。
- `-auth-sys-extended-groups` AUTH_SYSとともにを使用し、NFSv4.x ACLを使用しない場合は、NFS経由のアクセスを必要とするすべてのユーザが、ns-switchで指定されたネームサービスデータベース内に有効なUNIXユーザ（ローカルユーザも可）を必要とします。

`-v4-numeric-ids` オプションの詳細については、「名前文字列のバイパス–数値ID」を参照してください。

ローカルファイルの使用

`-auth-sys-extended-groups` オプションを使用すると16個を超える補助グループをサポートできます。このオプションは、LDAPなどの外部ネームサービスに適用するのが最適です。ただし、外部ネームサービスサーバを使用できない場合は、ローカルファイルを使用してでサポートするUNIXユーザおよびグループを作成できます `-auth-sys- extended-groups`。使用するユーザとグループを作成し、 `unix-group adduser` コマンドを実行して補助グループのリストを入力するだけです。

LDAP設定の詳細については、TR-4835：『How to Configure LDAP in ONTAP』を参照してください。

Active Directory LDAPに関する考慮事項

Microsoft Active Directory LDAPサーバでは、MaxPageSize属性がデフォルトの1、000に設定されています。これは、LDAPクエリで1、000を超えるグループが切り捨てられることを意味します。拡張グループに対して1、024の値が完全にサポートされるようにするには、MaxPageSize属性を1、024の値を反映するように変更する必要があります。この値を変更する方法については、Microsoft TechNetの「How to view and set LDAP policy in Active Directory by using Ntdsutil.exe」(英語)を参照してください。

この値の変更に関する懸念や、TechNetライブラリの記事 MaxPageSize is set too highを参照するには、Microsoftサポートにお問い合わせください。

フアイルスウノコウリヨシコウ

ファイル数の多い環境の詳細については、TR-4571：『NetApp FlexGroup Volume Best Practices and Implementation Guide』の対応するセクションを参照してください。

ファイル数の多い環境に推奨されるボリューム形式

ファイル数が多いということは、数十万から数十億個を超えるファイルを単一のネームスペースに作成して格納するワークロードを指します。これらのようなデータセットは多数のメタデータ処理を生成し、FlexVolボリュームのパフォーマンスに影響する可能性があります。ファイル数が多い環境では、FlexGroupボリュームを使用することを強く推奨します。FlexGroupボリュームの詳細については、TR-4571：『NetApp FlexGroup Volume Best Practices and Implementation Guide』を参照してください。

従来のオペレーティングシステムでのNFSの使用

次のセクションでは、新しいNFSプラットフォーム（WindowsやAppleオペレーティング システムなど）でのNFSの使用について説明します。Windows NFSのサポートは、ONTAP 8.2.3および8.3.1で初めて追加されました。

Windows向けNFS

ONTAP 8.2.3および8.3.1より前のONTAPシステムでNFSを使用する場合、サーバ管理者はHummingbirdやOpenText NFSクライアントなどのサードパーティ製ツールをインストールできます。Red Hatの Cygwinはは NFSをエミュレートしますが、利用するプロトコルはNFSではなくSMBであるため、CIFSライセンスが必要です。真のWindows NFSは、Services for Network File Systemまたは HummingbirdやOpenTextなどのサードパーティアプリケーションを通じてネイティブにのみ利用できます。

ONTAPのネイティブWindows NFS

RFC 1813では、次のセクションで、NFSでサポートされているクライアントとしてMS-DOSを取り上げています。

```
   nlm4_share

      struct nlm4_share
           { string    caller_name<LM_MAXSTRLEN>;
           netobj      fh;
           netobj      oh;
           fsh4_mode   mode;
           fsh4_access access;
      };

   This structure is used to support DOS file sharing. The
   caller_name field identifies the host making the request.
   The fh field identifies the file to be operated on. The oh
   field is an opaque object that identifies the host or process
   that is making the request. The mode and access fields
   specify the file-sharing and access modes. The encoding of fh
   is a byte count, followed by the file handle byte array. See
   the description of nlm4_lock for more details.
```

Windowsは、非監視対象のロック呼び出しとともにNLMを使用します。Windows NFSをサポートするには、次の非監視対象ロック呼び出しが必要です。

```
NLM_SHARE
NLM_UNSHARE
NLM_NM_LOCK
```

clustered ONTAP 8.3.1以前のバージョンやclustered ONTAP 8.2.3以前のバージョンでは、これらのロック呼び出しはサポートされていません。この点については、バグ296324で説明されています。

注： PCNFS、WebNFS、およびHCLNFS（従来のHummingbird NFSクライアント）はONTAPストレージシステムではサポートされておらず、これらのクライアントのサポートも予定されていません。

ONTAPでWindows NFSを使用する場合の考慮事項

ONTAPでWindows NFSを使用する場合は、次の考慮事項に注意してください。

- NSMはWindows NFSではサポートされていません。そのため、ボリューム移動やストレージフェイルオーバーを実行すると、NSMをサポートするNFSクライアントでは見られない原因の停止が発生する可能性があります。
- SVMでWindows NFSを使用している場合は、次のオプションを[Disabled]に設定する必要があります。EJUKEBOXが無効になっていない場合に発生する可能性のある問題の例は、バグ918755です。

```
enable-ejukebox
v3-connection-drop
```

> **注:** これらのオプションはデフォルトで有効になっています。無効にしてもNFSデータには影響しませんが、クライアントのクラッシュなど、予期しない動作が原因になる可能性があります。予期しない問題を回避するには、Windows NFSクライアント用に別のSVMを使用することを検討してください。

- Windows NFSは、必ずマウントオプションを使用してマウントし `mtype=hard` ます。
- Windows NFSを使用する場合は、showmountオプションを有効にする必要があります。そうしないと、Windows NFSクライアントでファイルやフォルダの名前変更が失敗することがあります。

```
cluster::> nfs server modify -vserver SVM -showmount enabled
```

- Windows NFSクライアントでは、 `df` コマンドで使用済みスペースと使用可能スペースを正しく表示できません。

WindowsでNFSをマウントする例：

```
C:\>mount -o mtype=hard \\x.x.x.e\unix Z:
Z: is now successfully connected to \\x.x.x.e\unix
```

```
The command completed successfully.
```

Windows NFSサポートの有効化

ONTAPには、Windows NFSクライアントを許可するために有効に切り替える必要のある特定のNFSサーバオプションがあります。このオプションはデフォルトで無効になっています。Windows NFSをサポートするONTAPバージョンからリバートする場合は、リバート前にこのオプションを無効にする必要があります。

```
cluster::> nfs server show -vserver nfs_svm -fields v3-ms-dos-client
vserver v3-ms-dos-client
---------- --------------------
nfs_svm disabled
```

Windows NFSのユースケース

Windows NFSは、Windowsオペレーティング システムで稼働するNetApp ストレージ デバイスへのアクセスを提供するために、CIFSライセンスの代わりに使用できます。それ以外には次のようなユースケースがあります。

- Windows上で実行され、NFS接続やLinux形式のコマンドと関数（GETATTRやSETATTRなど）を必要とするアプリケーション
- ユーザがCIFSではなくNFSプロトコルを利用する場合
- ユーザがマルチプロトコル接続を回避したい場合

Windowsを使用してNFS接続を提供することは可能ですが、パフォーマンスとNDOのためには、CIFSと、Windowsが提供する最新のSMBバージョンの新機能を使用する方が効果的です。また、ONTAPでWindows NFSを使用する場合は、後述する考慮事項を考慮する必要があります。

Apple OSを使用したNFS

NFSマウントは、Apple OSのFinderまたはターミナルウィンドウを使用して実行することもできます。Apple OSのすべてのマウントオプションについては、 `man mount_nfs` ターミナルウィンドウでコマンドを使用してください。NFSにAppleクライアントを使用する場合は、いくつかの点に留意する必要があります。

動的UIDと静的UID

MacとActive Directoryを併用した場合、デフォルトではユーザのWindows SIDを基にUID / GIDが動的に作成されます。通常はこれで十分ですが、UIDとGIDを制御する必要がある場合（既存のLDAPサーバに統合する場合など）は、静的なUIDを利用できます。Active DirectoryでApple OSを使用する場合のベストプラクティスについては、ホワイトペーパー『Integrate Active Directory Using Directory Using Directory Utility on Mac』を参照してください。

Apple OSはデフォルトでrootを無効にする

AppleのOSでは、デフォルトでrootユーザ（UID 0）が無効になっています。ユーザはroot以外のユーザ名でログインすることを求められ、rootレベルのタスクを実行する場合はsudoを使用する必要があります。rootユーザは再度有効にすることができます。

Apple UIDは501から始まります

AppleのUIDはUID 501から始まります。このUIDは、ONTAPのデフォルトのUNIXユーザではなく、ほとんどのネームサービスサーバにも存在しません。この状況は環境内のすべてのApple OSクライアントで同じであるため、同じUIDのユーザが複数存在する可能性があります。この状況に対処する方法は次のとおりです。

- クラスタまたはネーム サービス サーバにUID 501のユーザを作成して、すべてのAppleユーザを認証する。
- AppleでNFSを使用する各ユーザのUIDをApple OSで変更する。

NTFSセキュリティ形式のボリュームでのApple NFSの使用

Apple NFSでは、NTFSセキュリティ形式のボリュームの処理がLinux NFSクライアントとは異なります。そのため、NTFSセキュリティ形式が使用されている場合、Finderアプリケーションを使用したNFSマウントへのコピーや書き込みはデフォルトで失敗します。この問題は、Appleクライアントがそのファイルに対してEXCLUSIVE CREATE処理を試行したときに発生します。この処理は、ONTAP内のSMBクライアントでのみ許可されます。

回避策では、NFSサーバオプションを `-ntfs-unix-security-ops ignore`に設定することで、NTFSセキュリティ形式のボリュームがAppleのNFSマウントで適切に機能するようにすることができます。詳細については、バグ723115を参照してください。

NFSのrootonly処理がApple OSで想定どおりに機能しない

「rootonlyオプション-nfsrootonlyおよびmountrootonly」というセクションでは、rootonly操作について説明します。デフォルトでは `-mount-rootonly` 、は有効になっており、`-nfs-rootonly` は無効になっています。Apple OSでは、NFSを使用してマウントする場合、MOUNTプロトコルには予約ポートが、NFSプロトコルには非予約ポートが常に使用されます。LinuxのNFSマウント オプションresvport / noresvportはApple OSに適用されますが、noresvportではクラスタに送信されるクライアントのMOUNTポートが制御されません。したがって、Apple NFSクライアントは常に1024より小さいポートをMOUNTに使用します。

現時点では、この動作を変更する方法がわかっていないため、NFSマウント呼び出しに非予約ポートを使用するには、Appleのテクニカルサポートの協力が必要になります。NFSv4.xではMOUNTプロトコルを使用しないため、NFSv4.xマウントでは問題となりません。NFS処理用のNFSクライアント ポート（ポート2049）はresvport / noresvportマウント オプションで制御できますが、クラスタのNFSサーバ オプションはクライアントの動作に合わせて設定する必要があります。この操作はNFSのすべてのバージョンに影響します。

また、Apple OSでresvportオプションを指定してマウントする場合は、 `sudo` rootが無効になっており、 `-users` デフォルトではオプションが指定されていないため、コマンドを使用する必要があります。

注： Finderを使用してNFSをマウントする場合は、マウントオプションを指定できません。

付録A

次のセクションでは、コマンドの例、設定の出力、およびこのドキュメントの主要なセクションを混乱させる可能性のあるその他の情報について説明します。

例

次のセクションでは、コマンドとNFS機能の例を示します。

rootユーザの制御例

ストレージ管理者が、特定のクライアントからrootユーザの処理方法を制御したい場合があります。これらの例は、アプローチの仕組みを示しています。

rootノ引き下げ

次の各例では、さまざまな設定シナリオでrootの権限をanonに引き下げる方法を示します。

例1：すべてのクライアントのrootをanonユーザに引き下げます。

このアプローチでは、sec=sysを使用するすべてのNFSクライアントにスーパーユーザを使用します。それ以外のsecタイプはアクセスを拒否されます。

```
cluster::> vserver export-policy rule show -policyname root_squash -instance
  (vserver export-policy rule show)
```

```
                                    Vserver: vs0
                                Policy Name: root_squash
                                 Rule Index: 1
                            Access Protocol: nfs     ← only NFS is allowed (NFSv3 and v4)
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                             RO Access Rule: sys     ← only AUTH_SYS is allowed
                             RW Access Rule: sys     ← only AUTH_SYS is allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                   Superuser Security Types: none    ← superuser (root) squashed to anon user
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
---------- -------- ---------------
vs0      nfsvol root_squash

[root@nfsclient /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_squash

[root@nfsclient mnt]# ls -la
total 116
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nobody nobody      0 Apr 24 11:33 root_squash

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3      0     0 106496 Apr 24 2013 .
dr-xr-xr-x. 26     0     0   4096 Apr 24 11:24 ..
drwxrwxrwx. 12     0     0   4096 Apr 24 11:05 .snapshot
drwxr-xr-x. 2      0     1   4096 Apr 18 12:54 junction
-rw-r--r--.  1 65534 65534      0 Apr 24 2013 root_squash

[root@nfsclient /]# mount -o sec=krb5 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol
```

例2：特定のクライアントに対してsuperuserを使用してrootをanonユーザに引き下げます。

この例では、sec=sysおよびsec=noneを指定できます。

```
cluster::> vserver export-policy rule show -policyname root_squash_client -instance
  (vserver export-policy rule show)

                                    Vserver: vs0
                                Policy Name: root_squash_client
                                 Rule Index: 1
                            Access Protocol: nfs     ← only NFS is allowed (NFSv3 and v4)
Client Match Hostname, IP Address, Netgroup, or Domain: x.x.x.x     ← just this client
                             RO Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE are allowed
                             RW Access Rule: sys,none       ← AUTH_SYS and AUTH NONE are allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                   Superuser Security Types: none    ← superuser (root) squashed to anon user
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
---------- -------- ---------------
vs0      nfsvol root_squash_client

[root@nfsclient /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_squash_client

[root@nfsclient mnt]# ls -la
```

```
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nfsnobody nfsnobody      0 Apr 24 2013 root_squash_client

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3            0 106496 Apr 24 2013 .
dr-xr-xr-x. 26           0   4096 Apr 24 11:24 ..
drwxrwxrwx. 12           0   4096 Apr 24 11:05 .snapshot
drwxr-xr-x. 2            1   4096 Apr 18 12:54 junction
 -rw-r--r--. 1 65534 65534      0 Apr 24 2013 root_squash_client
```

例3：特定のクライアントセットに対してsuperuserを使用してrootをanonユーザに引き下げます。

このアプローチではsec=krb5（Kerberos）を使用し、NFSv4とCIFSのみが許可されます。

```
cluster::> vserver export-policy rule show -policyname root_squash_krb5 -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_squash_krb5
                             Rule Index: 1
                        Access Protocol: nfs4,cifs ← only NFSv4 and CIFS are allowed Client
Match Hostname, IP Address, Netgroup, or Domain: x.x.x.0/24 ← just clients with an IP address
of 10.10.100.X
                         RO Access Rule: krb5    ← only AUTH_RPCGSSD is allowed
                         RW Access Rule: krb5    ← only AUTH_RPCGSSD is allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: none    ← superuser (root) squashed to anon user
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
---------- -------- --------------
vs0      nfsvol root_squash

[root@nfsclient /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol

[root@nfsclient /]# mount -t nfs4 cluster:/nfsvol /mnt
mount.nfs4: Operation not permitted

[root@nfsclient /]# mount -t nfs4 -o sec=krb5 krbsn:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_squash_krb5

[root@nfsclient mnt]# ls -la
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nobody nobody      0 Apr 24 11:50 root_squash_krb5

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3 0 0 106496 Apr 24 2013 .
dr-xr-xr-x. 26 0 0     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 0 1     4096 Apr 18 12:54 junction
-rw-r--r--.  1 99 99      0 Apr 24 11:50 root_squash_krb5
```

注： この例のUID 99は、NFSv4でユーザ名をNFSv4ドメインにマッピングできない場合に発生します。を参照すると、 `/var/log/messages` 次のことが確認されます。

```
Apr 23 10:54:23 nfsclient nfsidmap[1810]: nss_getpwnam: name 'pcuser' not found in domain
nfsv4domain.netapp.com'
```

上記の各例では、rootユーザがあるマウントへのアクセスを要求すると、anon UIDにマッピングされます。この場合、UIDは65534です。これにより、指定したクライアントからNFS共有への不要なルート アクセスを防止できます。最初の2つの例では「sys」がrwおよびroアクセスルールとして指定されているため、 `sec=sys` Gainアクセスを使用しているクライアントのみが対象となります。3番目の例は、Kerberosに対応

したNFS認証を使用する設定を示しています。アクセス プロトコルをNFSに設定すると、共有へのNFSアクセスのみが許可されます（NFSv3とNFSv4を含む）。マルチプロトコル アクセスが必要である場合は、NFSとCIFSを許可するようにアクセス プロトコルを設定する必要があります。ここでも、NFSアクセスをNFSv3またはNFSv4のみに限定できます。

rootはroot（no_root_squash）

次の各例では、rootユーザがrootユーザとしてNFS共有にアクセスできるようにする方法を示します。この方法は、「no_root_squash」とも呼ばれます。

例1：sec=sysの場合にのみ、すべてのクライアントにsuperuserを使用してrootにrootとしてアクセスを許可します。

この例では、sec=noneおよびsec=sysにrwおよびroアクセスが許可され、その他のanonアクセスはすべて65534にマッピングされます。

```
cluster::> vserver export-policy rule show -policyname root_allow_anon_squash -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_allow_anon_squash
                             Rule Index: 1
                        Access Protocol: nfs     ← only NFS is allowed (NFSv3 and v4)
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                         RO Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE allowed
                         RW Access Rule: sys,none       ← AUTH_SYS and AUTH_NONE allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: sys     ← superuser for AUTH_SYS only
              Honor SetUID Bits in SETATTR: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
---------- -------- ---------------
vs0      nfsvol root_allow_anon_squash

[root@nfsclient /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_allow_anon_squash_nfsv3

[root@nfsclient mnt]# ls -la
drwxrwxrwx. 3 root       root      106496 Apr 24 2013 .
dr-xr-xr-x. 26 root      root        4096 Apr 24 11:24 ..
drwxrwxrwx. 12 root      root        4096 Apr 24 11:05 .snapshot
drwxr-xr-x. 2 root       bin         4096 Apr 18 12:54 junction
-rw-r--r--. 1 root       root           0 Apr 24 2013 root_allow_anon_squash_nfsv3

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3 0 0 106496 Apr 24 2013 .
dr-xr-xr-x. 26 0 0     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 0 1     4096 Apr 18 12:54 junction
-rw-r--r--.  1 0 0        0 Apr 24 11:56 root_allow_anon_squash_nfsv3
```

例2：sec=krb5の場合にのみ、superuserを使用してrootにrootとしてアクセスを許可します。

この例では、anonアクセスを65534にマッピングしています。sec=sysとsec=krb5が許可されますが、使用できるのはNFSv4だけです。

```
cluster::> vserver export-policy rule show -policyname root_allow_krb5_only -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_allow_krb5_only
                             Rule Index: 1
                        Access Protocol: nfs4    ← only NFSv4 is allowed
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
```

```
                            RO Access Rule: sys,krb5       ← AUTH_SYS and AUTH_RPCGSS allowed
                            RW Access Rule: sys,krb5       ← AUTH_SYS and AUTH_RPCGSS allowed
User ID To Which Anonymous Users Are Mapped: 65534   ← mapped to 65534
                  Superuser Security Types: krb5    ← superuser via AUTH_RPCGSS only
              Honor SetUID Bits in SETATTR: true

cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
---------- -------- --------------
vs0      nfsvol root_allow_krb5_only

[root@nfsclient /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol

[root@nfsclient /]# mount -t nfs4 cluster:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_allow_krb5_only_notkrb5

[root@nfsclient mnt]# ls -la
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--.  1 nobody nobody      0 Apr 24 2013 root_allow_krb5_only_notkrb5

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3 0 0 106496 Apr 24 2013 .
dr-xr-xr-x. 26 0 0     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 0 1     4096 Apr 18 12:54 junction
-rw-r--r--.  1 99 99      0 Apr 24 2013 root_allow_krb5_only_notkrb5


NOTE: Again, the UID of an unmapped user in NFSv4 is 99. This is controlled via /etc/idmapd.conf
in Linux and /etc/default/nfs in Solaris.

[root@nfsclient /]# mount -t nfs4 -o sec=krb5 cluster:/nfsvol /mnt
[root@nfsclient /]# kinit
Password for root@KRB5DOMAIN.NETAPP.COM:
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_allow_krb5_only_krb5mount

[root@nfsclient mnt]# ls -la
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--. 1 root    daemon      0 Apr 24 2013 root_allow_krb5_only_krb5mount

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3 0 0 106496 Apr 24 2013 .
dr-xr-xr-x. 26 0 0     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 0 1     4096 Apr 18 12:54 junction
-rw-r--r--.  1 0 1        0 Apr 24 2013 root_allow_krb5_only_krb5mount
```

例3：anon=0を使用してrootユーザとすべての匿名ユーザにrootとしてアクセスを許可します。

この例では、NFSv4でのsec=sysとsec=krb5のみを許可しています。

```
cluster::> vserver export-policy rule show -policyname root_allow_anon0 -instance
  (vserver export-policy rule show)

                                Vserver: vs0
                            Policy Name: root_allow_anon0
                             Rule Index: 1
                        Access Protocol: nfs4    ← only NFSv4 is allowed
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0   ← all clients
                            RO Access Rule: krb5, sys      ← AUTH_SYS and AUTH_RPCGSS allowed
                            RW Access Rule: krb5, sys      ← AUTH_SYS and AUTH_RPCGSS allowed
User ID To Which Anonymous Users Are Mapped: 0      ← mapped to 0
                  Superuser Security Types: none    ← superuser (root) squashed to anon user
```

```
               Honor SetUID Bits in SETATTR: true
                  Allow Creation of Devices: true


cluster::> volume show -vserver vs0 -volume nfsvol -fields policy
vserver volume policy
---------- -------- --------------
vs0      nfsvol root_allow_anon0

[root@nfsclient /]# mount -o nfsvers=3 cluster:/nfsvol /mnt
mount.nfs: access denied by server while mounting cluster:/nfsvol

[root@nfsclient /]# mount -t nfs4 cluster:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_allow_anon0

[root@nfsclient mnt]# ls -la
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--. 1 root    daemon      0 Apr 24 2013 root_allow_anon0

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3 0 0 106496 Apr 24 2013 .
dr-xr-xr-x. 26 0 0     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 0 1     4096 Apr 18 12:54 junction
-rw-r--r--.  1 0 1        0 Apr 24 2013 root_allow_anon0


[root@nfsclient /]# mount -t nfs4 -o sec=krb5 cluster:/nfsvol /mnt
[root@nfsclient /]# cd /mnt

[root@nfsclient mnt]# touch root_allow_anon0_krb5

[root@nfsclient mnt]# ls -la
drwxrwxrwx. 3 root    root   106496 Apr 24 2013 .
dr-xr-xr-x. 26 root   root     4096 Apr 24 11:24 ..
drwxr-xr-x. 2 root    daemon   4096 Apr 18 12:54 junction
-rw-r--r--. 1 root    daemon      0 Apr 24 2013 root_allow_anon0_krb5

[root@nfsclient mnt]# ls -lan
drwxrwxrwx. 3 0 0 106496 Apr 24 2013 .
dr-xr-xr-x. 26 0 0     4096 Apr 24 11:24 ..
drwxr-xr-x.  2 0 1     4096 Apr 18 12:54 junction
-rw-r--r--.  1 0 1        0 Apr 24 2013 root_allow_anon0_krb5
```

vsrootノフカキョウユウミラーノサクセイレイ

この例は、vsrootボリュームの負荷共有ミラーを作成する方法を示しています。負荷共有ミラーを使用するのは、環境NFSv3およびSMBのみです。

```
cluster::*> vol show -vserver NFS -vsroot true -fields size
vserver volume size
---------- -------- ------
NFS      vsroot 1GB

cluster::*> vol create -vserver NFS -volume vsroot_mirror1 -aggregate aggr1_node1 -size 1g -type
DP
[Job 26705] Job succeeded: Successful

cluster::*> vol create -vserver NFS -volume vsroot_mirror2 -aggregate aggr1_node2 -size 1g -type
DP
[Job 26707] Job succeeded: Successful

cluster::*> snapmirror create -source-path NFS:vsroot -destination-path NFS:vsroot_mirror1 -
vserver NFS -type LS -schedule daily
[Job 26709] Job succeeded: SnapMirror: done
```

```
cluster::*> snapmirror create -source-path NFS:vsroot -destination-path NFS:vsroot_mirror2 -
vserver NFS -type LS -schedule daily
[Job 26711] Job succeeded: SnapMirror: done

cluster::*> snapmirror show -source-path NFS:vsroot -fields schedule,state
source-path                   destination-path                        schedule state
----------------------------- --------------------------------------- -------- ------------
cluster://NFS/vsroot          cluster://NFS/vsroot_mirror1 daily      -        -
cluster://NFS/vsroot          cluster://NFS/vsroot_mirror2 daily      -        -
2 entries were displayed.

cluster::*> snapmirror initialize-ls-set -source-path NFS:vsroot
[Job 26714] Job is queued: snapmirror initialize-ls-set for source " cluster://NFS/vsroot".

cluster::*> snapmirror show -source-path NFS:vsroot -fields schedule,state
source-path                   destination-path                        schedule state
----------------------------- --------------------------------------- -------- ------------
cluster://NFS/vsroot          cluster://NFS/vsroot_mirror1            daily    Snapmirrored
cluster://NFS/vsroot          cluster://NFS/vsroot_mirror2            daily    Snapmirrored
2 entries were displayed.
```

エクスポートポリシールールの継承の例

次の例では、SVMルートボリュームのスーパーユーザアクセスがクライアントのみに制限されて `x.x.x.x` います。rootユーザが以外のクライアントからマウントにアクセスしようとする `x.x.x.x`と、rootユーザはanonユーザ（65534）に引き下げられます。

```
cluster::> vol show -vserver nfs_svm -volume rootvol -fields policy
  (volume show)
vserver volume policy
---------- ---------- ----------
nfs_svm rootvol default

cluster::> export-policy rule show -vserver nfs_svm -policyname default -instance
  (vserver export-policy rule show)

                                Vserver: nfs_svm
                            Policy Name: default
                             Rule Index: 1
                        Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: x.x.x.x
                          RO Access Rule: any
                          RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: any
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true

                                Vserver: nfs_svm
                            Policy Name: default
                             Rule Index: 2
                        Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                          RO Access Rule: any
                          RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: none
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
2 entries were displayed.
```

「SVMルートボリュームへのアクセスの制限」セクションの例と同様に、rootはボリューム権限（711）およびx.x.x.x以外のホストの既存のエクスポートポリシールールに基づいてSVMルートの内容をリストできません。

```
# ifconfig | grep "inet addr"
          inet addr:x.x.x.y Bcast:x.x.225.255 Mask:255.255.255.0
          inet addr:127.0.0.1 Mask:255.0.0.0
```

```
# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
# mount | grep mnt
x.x.x.e:/ on /mnt type nfs (rw,nfsvers=3,addr=x.x.x.e)
# cd /mnt
# ls
ls: cannot open directory .: Permission denied
```

SVM内のデータボリュームにもこのエクスポートポリシーが設定されている場合は、同じルールが使用され、ルートアクセスを許可するクライアントのみがrootとしてログインできます。

データ ボリュームへのルート アクセスが必要な場合は、新しいエクスポート ポリシーを作成し、サブネットやネットグループを指定して、あるいはクライアントIPアドレスまたはホスト名を個別に指定した複数のルールを指定して、すべてのホストまたは一部のホストにルート アクセスを指定できます。

他のエクスポート ポリシー ルール属性（RWなど）についても同様の考え方が当てはまります。

たとえばx.x.x.x、スーパーユーザを除くすべてのクライアントに書き込みアクセスを禁止するようにデフォルトのエクスポートポリシールールを変更すると、そのエクスポートポリシーが適用されているボリュームには、rootであっても書き込みアクセスが禁止されます。

```
cluster::> export-policy rule modify -vserver nfs_svm -policyname default -ruleindex 2 -rwrule
never -superuser any

cluster::> export-policy rule show -vserver nfs_svm -policyname default -instance
  (vserver export-policy rule show)

                                Vserver: nfs_svm
                            Policy Name: default
                             Rule Index: 1
                          Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: x.x.x.x
                           RO Access Rule: any
                           RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: any
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true

                                Vserver: nfs_svm
                            Policy Name: default
                             Rule Index: 2
                          Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                           RO Access Rule: any
                           RW Access Rule: never
User ID To Which Anonymous Users Are Mapped: 65534
                  Superuser Security Types: any
              Honor SetUID Bits in SETATTR: true
                 Allow Creation of Devices: true
2 entries were displayed.

# ifconfig | grep "inet addr"
          inet addr:x.x.x.y Bcast:x.x.225.255 Mask:255.255.255.0
          inet addr:127.0.0.1 Mask:255.0.0.0
# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
# mount | grep mnt
x.x.x.e:/ on /mnt type nfs (rw,nfsvers=3,addr=x.x.x.e)
# cd /mnt
# touch rootfile
touch: cannot touch `rootfile': Read-only file system
```

新しいポリシーとルールを作成してデータボリュームに適用すると、同じユーザに、SVMルートボリュームの下にマウントされているデータボリュームへの書き込みが許可されます。これは、SVMルート ボリュームのエクスポート ポリシー ルールで書き込みアクセスが禁止されていても同様です。

例：

```
cluster::> export-policy create -vserver nfs_svm -policyname volume
cluster::> export-policy rule create -vserver nfs_svm -policyname volume -clientmatch 0.0.0.0/0 -
rorule any -rwrule any -allow-suid true -allow-dev true -ntfs-unix-security-ops fail -chown-mode
restricted -superuser any -protocol any -ruleindex 1 -anon 65534

cluster::> export-policy rule show -vserver nfs_svm -policyname volume -instance
  (vserver export-policy rule show)

                                    Vserver: nfs_svm
                                Policy Name: volume
                                 Rule Index: 1
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                             RO Access Rule: any
                             RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 65534
                    Superuser Security Types: any
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true

::> volume modify -vserver flexvol -volume unix -policy volume
```

クライアントからの操作：

```
# ifconfig | grep "inet addr"
          inet addr:x.x.x.y Bcast:x.x.225.255 Mask:255.255.255.0
          inet addr:127.0.0.1 Mask:255.0.0.0
# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-
s0:c0.c1023
# cd /mnt/unix
[root@linux-client unix]# ls
file
[root@linux-client unix]# touch rootfile
[root@linux-client unix]# ls -la | grep rootfile
-rw-r--r--. 1 root root     0 Apr 1  2014 rootfile
# cd ..
# ls
nfs4 ntfs  unix
# touch rootdir
touch: cannot touch `rootdir': Read-only file system
```

ただし、エクスポートポリシールールの読み取り専用属性では、親からの読み取りアクセスを許可してマウントを実行する必要があります。`rorule never` 親ボリュームのエクスポートポリシー（空のポリシー）でエクスポートポリシールールを設定するかどうかを設定すると、その親の下にあるボリュームへのマウントが禁止されます。

次の例では `rorule`、vsrootボリュームのエクスポートポリシーがおよび `rwrule` に設定され `never`、データボリュームのエクスポートポリシーにはワイドオープンルールが設定されています。

```
cluster::> export-policy rule show -vserver nfs -policyname wideopen -instance
  (vserver export-policy rule show)

                                    Vserver: nfs
                                Policy Name: wideopen
                                 Rule Index: 1
                            Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                             RO Access Rule: any
                             RW Access Rule: any
User ID To Which Anonymous Users Are Mapped: 0
                    Superuser Security Types: any
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true

cluster::> export-policy rule show -vserver nfs -policyname deny -instance
  (vserver export-policy rule show)
```

```
                                    Vserver: nfs
                                Policy Name: deny
                                 Rule Index: 1
                             Access Protocol: any
Client Match Hostname, IP Address, Netgroup, or Domain: 0.0.0.0/0
                              RO Access Rule: never
                              RW Access Rule: never
User ID To Which Anonymous Users Are Mapped: 65534
                    Superuser Security Types: sys
                Honor SetUID Bits in SETATTR: true
                   Allow Creation of Devices: true

cluster::> volume show -vserver nfs -volume rootvol -fields policy,unix-permissions
vserver volume policy unix-permissions
---------- ---------- -------- ----------------------

nfs

cluster::> volume show -vserver nfs -volume unix -fields policy,unix-permissions
vserver volume policy   unix-permissions
---------- -------- ----------- ----------------------

nfs     unix   wideopen ---rwxrwxrwx
```

ボリュームunixのマウントを試行すると、アクセスは拒否されます。

```
# mount -o nfsvers=3 x.x.x.e:/unix /cdot
mount.nfs: access denied by server while mounting x.x.x.e:/unix
```

拒否ポリシーを変更して読み取り専用アクセスを許可すると、マウントが許可されます。

```
cluster::> export-policy rule modify -vserver nfs -policyname deny -rorule any -ruleindex 1

# mount -o nfsvers=3 x.x.x.e:/unix /cdot
# mount | grep unix
x.x.x.e:/unix on /cdot type nfs (rw,nfsvers=3,addr=x.x.x.e)
```

つまり、ストレージ管理者は、エクスポート ポリシー、ルール、およびボリューム権限を使用して、どのユーザがファイルシステムを表示してアクセスできるかを完全に、かつ細かく制御することができます。

エクスポートポリシールールインデックスの例

次の例では、IPアドレス x.x.x.x（ホスト名が centos64）のクライアントがボリュームの読み取りアクセスを拒否され、他のすべてのクライアントがアクセスを許可されています。ただし、クライアントルールは[すべてのアクセス]ルールの下にあるため、マウントと読み取りが許可されます。

例：

```
cluster::> export-policy rule show -vserver NAS -policyname allow_all
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
NAS          allow_all       1       any      0.0.0.0/0             any
NAS          allow_all       99      any      x.x.x.x        never
2 entries were displayed.

cluster::> vol show -vserver NAS -volume unix -fields policy
vserver volume policy
---------- -------- ------------
NAS     unix   allow_all

[root@centos64 ~]# mount -o nfsvers=3 x.x.x.a:/vol/nfs /7mode
[root@centos64 ~]# mount x.x.x.a:/unix /mnt
[root@centos64 ~]# cd /mnt
[root@centos64 mnt]# ls -la
total 12
drwxrwxrwx.  3 root root    4096 Dec 10 14:49 .
dr-xr-xr-x. 46 root root    4096 Dec 10 14:57 ..
drwxrwx---. 2 root root 4096 Dec 10 15:00 file
```

これらのルールが反転されると、ポリシー内のすべてのユーザにアクセスを許可するルールにもかかわらず、クライアントはアクセスを拒否されます。ルールインデックス番号は、 `export-policy rule setindex` コマンドを使用して変更できます。次の例では、ルール#1がルール#99に変更されています。ルール番号99はデフォルトでルール番号98に移動されます。

```
cluster::> export-policy rule setindex -vserver NAS -policyname allow_all -ruleindex 1 -
newruleindex 99

cluster::> export-policy rule show -vserver NAS -policyname allow_all
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
NAS          allow_all       98      any      x.x.x.x        never
NAS          allow_all       99      any      0.0.0.0/0             any
2 entries were displayed.

cluster::> export-policy cache flush -vserver NAS -cache all

Warning: You are about to flush the "all (but showmount)" cache for Vserver "NAS" on node
"node2", which will result in increased traffic to the name servers. Do you want to proceed with
         flushing the cache? {y|n}: y


[root@centos64 /]# mount x.x.x.a:/unix /mnt
mount.nfs: access denied by server while mounting x.x.x.a:/unix
```

NFSv4.x ACLの明示的拒否の例

たとえば、ユーザ `ldapuser` がグループDomain Usersに属しているとします。

```
sh-4.1$ id
uid=55(ldapuser) gid=513(Domain Users) groups=513(Domain Users),503(unixadmins)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

ボリュームに対する権限 `mixed` は775です。所有者はroot、グループはDomain Usersです。

```
[root@nfsclient /]# nfs4_getfacl /mixed
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A:g:GROUP@:rwaDxtTnNcy
D:g:GROUP@:C
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC

[root@nfsclient /]# ls -la | grep mixed
drwxrwxr-x.   3 root     Domain Users   4096 Apr 30 09:52 mixed
```

`ldapuser` はDomain Usersのメンバーであるため、ボリュームへの書き込みアクセス権を持つ必要があります。この権限は次のとおりです。

```
sh-4.1$ cd /mixed
sh-4.1$ ls -la
total 12
drwxrwxr-x. 3 root Domain Users 4096 Apr 30 09:52 .
dr-xr-xr-x. 28 root root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root root         4096 Apr 30 08:00 .snapshot
sh-4.1$ touch newfile
sh-4.1$ nfs4_getfacl /mixed
sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 09:56 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile
```

ただし、ACLの順序が変更され、Everyoneに対する明示的なdenyがグループの前に配置され `ldapuser` た場合は、その書き込み先と同じボリュームへの書き込みアクセスが拒否されます。

```
[root@nfsclient /]# nfs4_getfacl /mixed
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A::EVERYONE@:rxtncy
D::EVERYONE@:waDTC
A:g:GROUP@:rwaDxtTnNcy

[root@nfsclient /]# su
ldapuser sh-4.1$ cd /mixed
sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 09:56 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile

sh-4.1$ touch newfile2
touch: cannot touch `newfile2': Permission denied
```

明示的な拒否ルールを削除すると、必要なアクセスがリストアされます。

```
[root@nfsclient /]# nfs4_getfacl /mixed
A::OWNER@:rwaDxtTnNcCy
D::OWNER@:
A::EVERYONE@:rxtncy
A:g:GROUP@:rwaDxtTnNcy

[root@nfsclient /]# su ldapuser

sh-4.1$ cd /mixed

sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 09:56 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile

sh-4.1$ touch newfile2

sh-4.1$ ls -la
total 12
drwxrwxr-x.  3 root     Domain Users 4096 Apr 30 10:06 .
dr-xr-xr-x. 28 root     root         4096 Apr 29 15:24 ..
drwxrwxrwx.  6 root     root         4096 Apr 30 08:00 .snapshot
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 09:56 newfile
-rw-r--r--.  1 ldapuser Domain Users    0 Apr 30 10:06 newfile2
```

NFSv4.xのACL保持の例

これは、新しく作成されたUNIX形式のボリュームです。

```
cluster::> volume show -vserver vs0 -volume unix -fields security-style,
unix-permissions,user,group
vserver volume user group security-style unix-permissions
---------- -------- ------ -------- -------------------- ----------------------
vs0     unix   0    1     unix           ---rwxr-xr-x

cluster::> vserver security file-directory show -vserver vs0 -path /unix

                Vserver: vs0
              File Path: /unix
      Security Style: unix
     Effective Style: unix
      DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
        Unix User Id: 0
       Unix Group Id: 1
```

```
         Unix Mode Bits: 755
 Unix Mode Bits in Text: rwxr-xr-x
                   ACLs: -
```

上記の例では、ボリューム（/unix）に755権限が設定されています。これは、所有者にはすべてのアクセスがあり、所有元グループには読み取り / 実行アクセスがあり、それ以外の全員に読み取り / 実行アクセスがあることを意味します。

fsecurityの出力にNFSv4 ACLは表示されませんが、クライアントから表示できるデフォルト値が設定されています。

```
[root@nfsclient /]# mount -t nfs4 krbsn:/unix /unix
[root@nfsclient /]# ls -la | grep unix
drwxr-xr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@nfsclient /]# nfs4_getfacl /unix
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy
```

前述のNFSv4 ACLは同じで、所有者にはすべてのアクセス権があり、所有者グループには読み取り/実行アクセス権があり、それ以外のすべてのACLには読み取り/実行アクセス権があります。デフォルトのモード ビットがNFSv4 ACLに関連付けられています。

モードビットを変更すると、NFSv4 ACLも変更されます。

```
[root@nfsclient /]# chmod 775 /unix
[root@nfsclient /]# ls -la | grep unix
drwxrwxr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@nfsclient /]# nfs4_getfacl /unix
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rxtncy
```

ACLにユーザACEを追加すると、そのエントリがアプライアンス上のACLに反映されます。さらに、ACL全体が自動的に読み込まれます。ACLはSID形式であることに注意してください。

```
[root@nfsclient /]# nfs4_setfacl -a A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy /unix
[root@nfsclient /]# nfs4_getfacl /unix
A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rxtncy

cluster::> vserver security file-directory show -vserver vs0 -path /unix

               Vserver: vs0
             File Path: /unix
         Security Style: unix
        Effective Style: unix
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 1
         Unix Mode Bits: 775
 Unix Mode Bits in Text: rwxrwxr-x
                   ACLs: NFSV4 Security Descriptor
                         Control:0x8014
                         DACL - ACEs
                           ALLOW-S-1-8-55-0x16019d
                           ALLOW-S-1-520-0-0x1601ff
                           ALLOW-S-1-520-1-0x1201ff-IG
                           ALLOW-S-1-520-2-0x1200a9
```

変換されたACLを表示するには、ボリュームを所有するノードでノードシェルからfsecurityを使用します。

```
cluster::> node run -node node2 fsecurity show /vol/unix
```

```
[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0775 (rwxrwxr-x)

  NFSv4 security descriptor:
    DACL:
      Allow - uid: 55 - 0x0016019d
      Allow - OWNER@ - 0x001601ff
      Allow - GROUP@ - 0x001201ff
      Allow - EVERYONE@ - 0x001200a9 (Read and Execute)
    SACL:
      No entries.
```

NFSv4 ACLが存在するときにモードビットを変更すると、設定したばかりのNFSv4 ACLがデフォルトで消去されます。

```
[root@nfsclient /]# chmod 755 /unix
[root@nfsclient /]# ls -la | grep unix
drwxr-xr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@nfsclient /]# nfs4_getfacl /unix
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy

cluster::> node run -node node2 fsecurity show /vol/unix

[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0755 (rwxr-xr-x)

  No security descriptor available.
```

ONTAPでこの動作を制御するには、次のdiagレベルオプションを使用します。

```
cluster::> set diag
cluster::*> nfs server modify -vserver vs0 -v4-acl-preserve [enabled|disabled]
```

このオプションをイネーブルにすると、モードビットが設定されてもACLはそのまま維持されます。

```
[root@nfsclient /]# nfs4_setfacl -a A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy /unix
[root@nfsclient /]# ls -la | grep unix
drwxr-xr-x.   2 root     daemon         4096 Apr 30 11:24 unix
[root@nfsclient /]# nfs4_getfacl /unix
A::ldapuser@nfsv4domain.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rxtncy
A::EVERYONE@:rxtncy

cluster::> vserver security file-directory show -vserver vs0 -path /unix

                Vserver: vs0
              File Path: /unix
        Security Style: unix
       Effective Style: unix
```

```
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 1
         Unix Mode Bits: 755
 Unix Mode Bits in Text: rwxr-xr-x
                   ACLs: NFSV4 Security Descriptor
                         Control:0x8014
                         DACL - ACEs
                           ALLOW-S-1-8-55-0x16019d
                           ALLOW-S-1-520-0-0x1601ff
                           ALLOW-S-1-520-1-0x1200a9-IG
                           ALLOW-S-1-520-2-0x1200a9

cluster::> node run -node node2 fsecurity show /vol/unix

[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0755 (rwxr-xr-x)

  NFSv4 security descriptor:
    DACL:
      Allow - uid: 55 - 0x0016019d
      Allow - OWNER@ - 0x001601ff
      Allow - GROUP@ - 0x001200a9 (Read and Execute)
      Allow - EVERYONE@ - 0x001200a9 (Read and Execute)
    SACL:
      No entries.
```

モードビットを設定してもACLはそのまま維持されることに注意してください。

```
[root@nfsclient /]# chmod 777 /unix
[root@nfsclient /]# ls -la | grep unix
drwxrwxrwx.   2 root     daemon         4096 Apr 30 11:24 unix
[root@nfsclient /]# nfs4_getfacl /unix
A::ldapuser@win2k8.ngslabs.netapp.com:ratTnNcCy
A::OWNER@:rwaDxtTnNcCy
A:g:GROUP@:rwaDxtTnNcy
A::EVERYONE@:rwaDxtTnNcy

cluster::> vserver security file-directory show -vserver vs0 -path /unix

               Vserver: vs0
             File Path: /unix
         Security Style: unix
        Effective Style: unix
         DOS Attributes: 10
 DOS Attributes in Text: ----D---
Expanded Dos Attributes: -
           Unix User Id: 0
          Unix Group Id: 1
         Unix Mode Bits: 777
 Unix Mode Bits in Text: rwxrwxrwx
                   ACLs: NFSV4 Security Descriptor
                         Control:0x8014
                         DACL - ACEs
                           ALLOW-S-1-8-55-0x16019d
                           ALLOW-S-1-520-0-0x1601ff
                           ALLOW-S-1-520-1-0x1201ff-IG
                           ALLOW-S-1-520-2-0x1201ff

cluster::*> node run -node node2 fsecurity show /vol/unix
```

```
[/vol/unix - Directory (inum 64)]
  Security style: Unix
  Effective style: Unix

  DOS attributes: 0x0010 (----D---)

  Unix security:
    uid: 0
    gid: 1
    mode: 0777 (rwxrwxrwx)

  NFSv4 security descriptor:
    DACL:
      Allow - uid: 55 - 0x0016019d
      Allow - OWNER@ - 0x001601ff
      Allow - GROUP@ - 0x001201ff
      Allow - EVERYONE@ - 0x001201ff
    SACL:
      No entries.
```

NFS監査イベントの例

次の例は、NFSv3を使用した監査イベントの出力例です。

```
- <Event>
- <System>
  <Provider Name="Netapp-Security-Auditing" />
  <EventID>4663</EventID>
  <EventName>Get Object Attributes</EventName>
  <Version>1</Version>
  <Source>NFSv3</Source>
  <Level>0</Level>
  <Opcode>0</Opcode>
  <Keywords>0x8020000000000000</Keywords>
  <Result>Audit Success</Result>
  <TimeCreated SystemTime="2013-08-08T20:36:05.011243000Z" />
  <Correlation />
  <Channel>Security</Channel>
  <Computer>e284de25-3edc-11e2-92d0-123478563412/525c9a2c-dce2-11e2-b94f-123478563412</Computer>
  <Security />
  </System>
- <EventData>
  <Data Name="SubjectIP" IPVersion="4">x.x.x.x</Data>
  <Data Name="SubjectUnix" Uid="10000" Gid="503" Local="false" />
  <Data Name="ObjectServer">Security</Data>
  <Data Name="ObjectType">Directory</Data>
  <Data Name="HandleID">00000000000453;00;00000040;3a2cada4</Data>
  <Data Name="ObjectName">/</Data>
  <Data Name="InformationRequested">File Type; File Size; Last Accessed Time; Last Metadata
Modfied Time; Last Modified Time; Unix Mode; Unix Owner; Unix Group;</Data>
  </EventData>
  </Event>
```

NFSv4.xリファーラルの例

データボリュームはnode1に存在します。

```
cluster::> volume show -vserver vs0 -volume nfsvol -fields node
vserver volume node
---------- -------- ----------------
vs0     nfsvol node1
```

データLIFはnode2に存在します。

```
cluster::> net int show -vserver vs0 -lif data2 -fields curr-node,home-node
  (network interface show)
vserver lif   home-node     curr-node     address
```

```
----------  -------  ------------------ ------------------ ---------------
vs0      data2 node2          node2          x.x.x.a
```

データLIFはnode1にもあります。

```
cluster::> net int show -vserver vs0 -curr-node node1 -role data
  (network interface show)
            Logical    Status     Network            Current       Current Is
Vserver     Interface Admin/Oper  Address/Mask       Node          Port    Home
----------- ---------- ---------- ------------------ ------------- ------- ----
vs0
            data1         up/up    x.x.x.a/24     node1         e0a     true
```

クライアントは、IPアドレスでnode2のデータLIFにマウント要求を行います `x.x.x.a`。

```
[root@nfsclient /]# mount -t nfs4 x.x.x.a:/nfsvol /mnt
```

マウント場所は、クライアントによって指定されたIPアドレスになります。

```
[root@nfsclient /]# mount | grep /mnt
x.x.x.a:/nfsvol on /mnt type nfs4 (rw,addr=x.x.x.a,clientaddr=x.x.x.z)
```

ただし、クラスタからは、実際にはデータボリュームが存在するnode1への接続が確立されたことがわかります。ノード2への接続は確立されていません。

```
cluster::> network connections active show -node node1 -service nfs*
             Vserver   Interface         Remote
      CID Ctx Name     Name:Local Port   Host:Port             Protocol/Service
----------- ---------- ----------------- ----------------------------------------
Node: node1
286571835   6 vs0          data:2049         x.x.x.z:763    TCP/nfs

cluster::> network connections active show -node node2 -service nfs*
There are no entries matching your query.
```

nfs connected-clientsの出力例

次に、nfs connected-clientsコマンドの出力例を示します。

```
cluster::*> nfs connected-clients show -vserver DEMO

     Node: node1
  Vserver: DEMO
  Data-Ip: 10.x.x.a
Client-Ip       Volume-Name       Protocol Idle-Time     Local-Reqs Remote-Reqs
--------------- ----------------- -------- ------------- ---------- -----------
10.x.x.b        scripts           nfs3     1h 3m 46s     391        0
10.x.x.c        XCP_catalog       nfs3     17s           0          372
10.x.x.d        scripts           nfs3     17s           372        0
10.x.x.c        home              nfs4     12h 50m 28s   256        0
10.x.x.b        home              nfs4.1   9h 51m 48s    47         0
```

パフォーマンスの比較：nconnectとpNFSを使用したNFSv3とNFSv4

当社のCPOCラボでは、NetApp製品の潜在的なパフォーマンスと機能のメリットを示すために、制御されたラボ環境で実際のワークロードを実行します。

このようなCPOCラボでは、次のコンポーネントを使用してシーケンシャルリードのワークロードをテストしました。

- NVIDIA DGX-2クライアント×1
- Ubuntu 20.04.2
- pNFSおよびnconnectを使用したNFSv4.1
- AFF A400クラスタ

- NetApp FlexGroupボリューム
- 256K wsize/rsize
- 100GbE接続
- 1GBファイル×32

その結果,以下の結果が得られた。レイテンシはどちらも1ミリ秒未満でした。

表32）NFSv4.1 / pNFS / nconnectとNFSv3–シーケンシャルリード：

テスト	帯域幅
NFSv3	10.2 GBps
NFSv4.1 / pNFS	21.9GBps

注： NFSv3とNFSv4.1では、どちらもnConnect=16を使用します。

このテストでは、pNFSを使用したNFSv4.1で、シーケンシャル読み取りワークロードのパフォーマンスが250usのレイテンシで2倍になりました。ファイルのサイズは1GBであるため、読み取りはほぼすべてコントローラのRAMから行われました。

シーケンシャルワークロードには、次のようなものがあります。

- Hadoop
- VMware
- データベースワークロード
- ハイパフォーマンスコンピューティングワークロード
- メディアワークロード

シーケンシャルワークロードの場合は、パフォーマンスを最大限に高めるためにNFSv4.1、pNFS、nconnectを検討してください。getattrsや大量のメタデータ（EDAやソフトウェアのビルドなど）を含むワークロードでは、これらのタイプのワークロードほどパフォーマンスが向上することはないことに注意してください。CPOCラボの詳細については、NetApp営業チームにお問い合わせください。

注： Oracleワークロードにも NFSv4.1とpNFSのメリットがありますが、全体的な結果を得るには、nconnectではなくdNFSを使用してください。詳細については、TR-3633：『Oracle Databases on ONTAP』を参照してください。また、Tech OnTapポッドキャストのエピソード279：『NetApp and Oracle‐What's New』もご覧ください。

NFSv3とNFSv4.x -パフォーマンスの比較

環境内のNFSv3とNFSv4.xのパフォーマンスへの影響を正確に把握するには、テストが不可欠です。すべてのワークロードが各プロトコルで同じ処理を実行するわけではありません。NFSv4.xは、ロック、状態フル、メタデータ処理、複合処理など、さまざまな点でNFSv3と大きく異なります。NFSv3のワークロードのプロファイルは、NFSv4.xを使用する同じワークロードとはまったく異なる場合があります。たとえば、ONTAP 9.1 NFSv3を使用した大量のファイル作成ワークロード（1、000ディレクトリ、それらのディレクトリ全体で100万ファイル）では、作成、検索、書き込みがほぼ均等に分割されたこのような統計が生成されます。

```
Object: nfsv3
Instance: DEMO
    Counter                                                     Value
    ---------------------------------------- ----------------------------------------
    access_total                                                 1014
    create_percent                                                33%
    create_total                                               1000000
    fsinfo_total                                                    4
    getattr_total                                                   3
    lookup_percent                                                33%
    lookup_total                                               1000003
    mkdir_total                                                  1003
    null_total                                                      4
    pathconf_total                                                  2
    total_ops                                                   31936
```

```
    write_percent                                          33%
    write_total                                       1000000
```

NFSv4.1を使用した場合とまったく同じワークロードでは、統計プロファイルがまったく異なります。NFSv4.xでのファイル作成時の動作がNFSv3と大きく異なるため、次のようになります。

- メタデータが大きい（15%）
- 作成処理の数が減少（1、000対1000000（NFSv3使用時）
- ファイル作成ごとのオープン/クローズ操作
- その他のメタデータ処理タイプ（COMPOUND、SEQUENCE、GETFH、PUTFH）
- 書き込みの合計は同じですが、パーセンテージははるかに低くなります（7%）。

```
Object: nfsv4_1
Instance: DEMO
    Counter                                                Value
    ---------------------------------------- ----------------------------------------
    access_percent                                          7%
    access_total                                       1001014
    close_percent                                           7%
    close_total                                        1000000
    compound_percent                                       23%
    compound_total                                     3002078
    create_session_total                                     2
    create_total                                          1003
    exchange_id_total                                        8
    getattr_percent                                        15%
    getattr_total                                      2002064
    getfh_percent                                           7%
    getfh_total                                        1001012
    lookup_total                                             7
    open_percent                                            7%
    open_total                                         1000000
    putfh_percent                                          23%
    putfh_total                                        3002064
    putrootfh_total                                          2
    reclaim_complete_total                                   2
    sequence_percent                                       23%
    sequence_total                                     3002068
    total_ops                                             6417
    write_percent                                           7%
    write_total                                        1000000
```

そのため、このワークロードにNFSv4.1を使用するだけで全体的なメタデータ処理が増え、NFSv4.1ではパフォーマンスが低下する傾向があります。表33に示すように、さまざまなパフォーマンス指標を並べて比較します。

表33）NFSv3とNFSv4.1のパフォーマンス–ファイル作成ワークロードが高い

テスト	平均 IOPS	平均MBps	平均レイテンシ	完了時間	平均CPU利用率
NFSv3	55118	71.9	1.6ミリ秒	54.7秒	51%
NFSv4.1	25068	13.9	11.5ミリ秒	283.5秒	24%

ご覧のように、NFSv4.1を使用している場合、このワークロードのレイテンシは高く、IOPSとスループットは低く、完了時間は約5倍になります。ストレージにそれほど多くの処理（IOPSの処理など）が要求されていないため、CPU使用率が低下します。

ただし、NFSv4.xのパフォーマンスが常にNFSv3よりも劣るわけではありません。NFSv4.xのパフォーマンスが同等以上になる場合もあります。これらはすべて、使用するワークロードのタイプによって異なります。

シーケンシャル性の高い書き込みワークロードの場合、NFSv4.1では処理するメタデータが少ないため、NFSv3との競合が発生します。マルチスレッドのdd処理を使用して8つの10GBファイルを作成するのは、NFSv3ワークロードのプロファイルです。

```
Object: nfsv3
Instance: DEMO
    Counter                                                    Value
---------------------------------------------------- ----------------------------------------
    access_total                                                  18
    create_total                                                   8
    fsinfo_total                                                   4
    getattr_total                                                  7
    lookup_total                                                  11
    mkdir_total                                                   11
    null_total                                                     4
    pathconf_total                                                 2
    total_ops                                                   5357
    write_percent                                                 99%
    write_total                                                1248306
```

この場合、書き込みはほぼ100%になります。NFSv4.1では書き込みの割合は低くなりますが、付随するメタデータ処理によってパフォーマンスが低下するタイプ（COMPOUNDおよびPUTFH）ではありません。

```
Object: nfsv4_1
Instance: DEMO
    Counter                                                    Value
    ---------------------------------------------------- ----------------------------------------
    access_total                                                  25
    close_total                                                    4
    compound_percent                                              33%
    compound_total                                             1238160
    create_session_total                                           4
    create_total                                                  11
    destroy_clientid_total                                         3
    destroy_session_total                                          3
    exchange_id_total                                             16
    getattr_total                                                 72
    getdeviceinfo_total                                            8
    getfh_total                                                   28
    layoutget_total                                                8
    layoutreturn_total                                             1
    lookup_total                                                   7
    open_total                                                     8
    putfh_percent                                                 33%
    putfh_total                                                1238107
    putrootfh_total                                                2
    reclaim_complete_total                                         4
    sequence_percent                                              33%
    sequence_total                                             1238134
    total_ops                                                    15285
    write_percent                                                 33%
    write_total                                                1238032
```

これにより、NFSv4.1のパフォーマンスとの比較が大幅に向上します。表34に示すように、IOPS、スループット、レイテンシはNFSv3とほぼ同じです。NFSv4.1はNFSv3よりも8秒長く（約3.7%多く）かかります。

表34）NFSv3とNFSv4.1のパフォーマンス-シーケンシャルライトが多い

テスト	平均 IOPS	平均MBps	平均レイテンシ	完了時間	平均CPU利用率
NFSv3	6085	383.1	5.4ミリ秒	216.6秒	28%
NFSv4.1	6259	366.3	4.4	224.7秒	26%

より標準的なベンチマークツール（vdbench）を使用すると、ワークロードタイプごとに読み取りパフォーマンスと書き込みパフォーマンスの違いを確認できます。ほとんどの場合、NFSv3の方が優れたパフォーマンスを発揮しますが、既存のデータセットを読み書きするワークロードでは、この差が特に大きいわけではありません。書き込みの方がパフォーマンスの差が大きい傾向がありますが、読み取りのパフォーマンスはほぼ同じです。テストでは、次の図に示すように、シーケンシャルライトのパフォーマンスもNFSv4.xで若干向上しました。

図23）ランダムリード、4K、NFSv3とNFSv4.xの比較- IOPS/レイテンシ

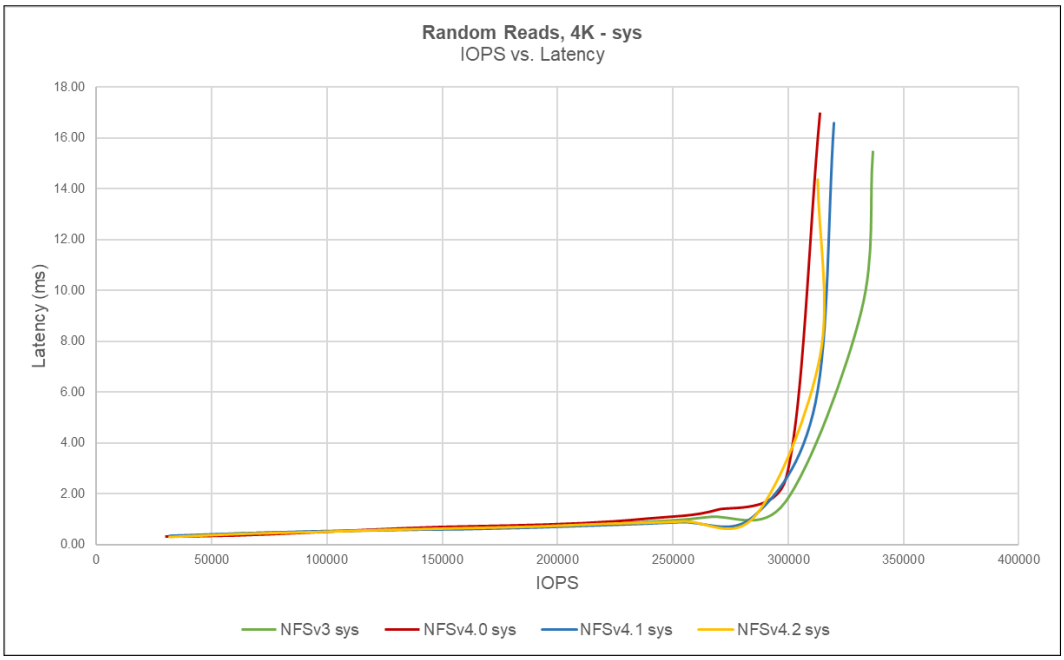


図24）ランダムライト（4K、NFSv3とNFSv4.xの比較- IOPS/レイテンシ

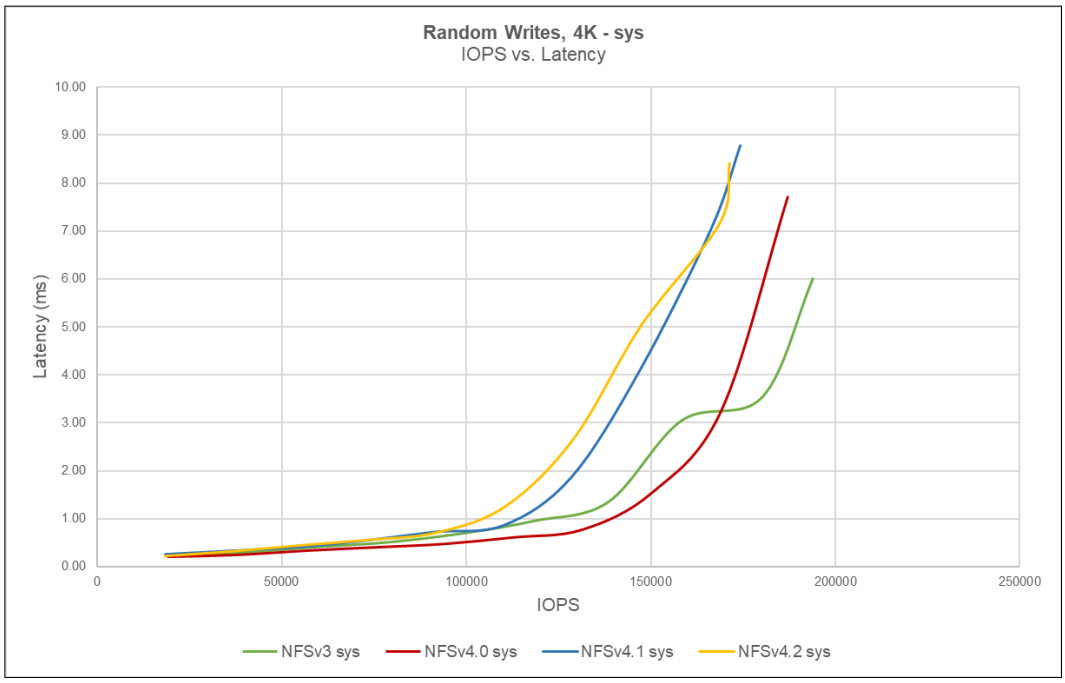


図25）シーケンシャル読み取り、32K、NFSv3とNFSv4.xの比較- IOPS/レイテンシ

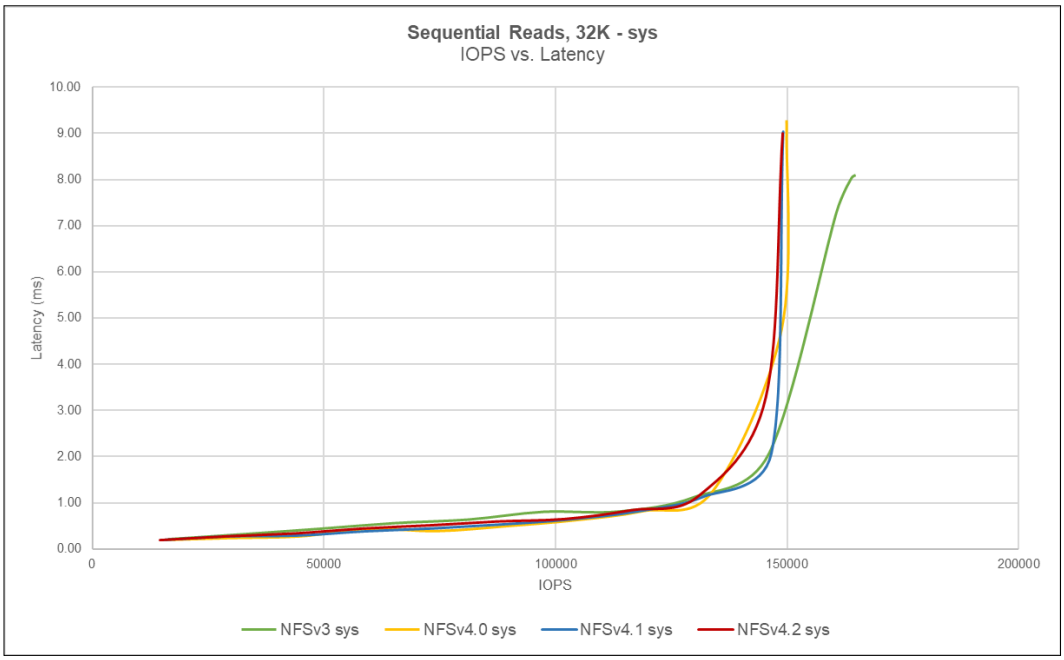
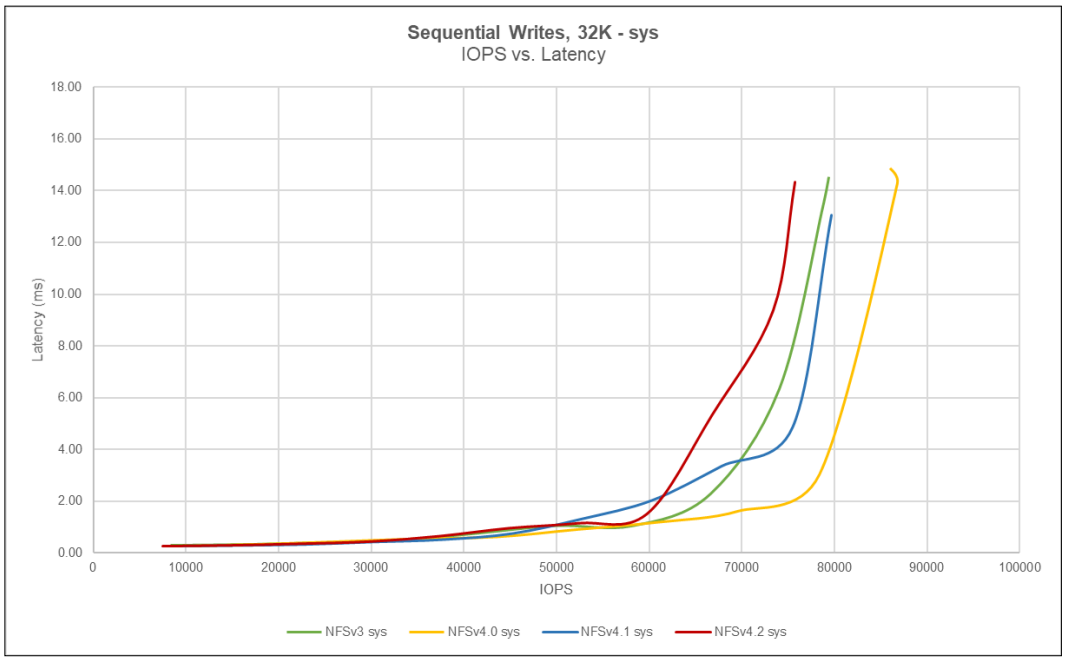


図26）シーケンシャルライト（32K、NFSv3とNFSv4.xの比較- IOPS/レイテンシ



TCPサイタイテンソウウイントウサイスノハフオオマンスレイ

次に、マルチスレッドファイルおよびディレクトリの作成を使用した次のパフォーマンステストの例を示します。

- ファイル数が多い（100万個のファイル-ディレクトリ1、000個、ディレクトリ1、000個の4KBファイル）
- 少ないファイル数で大容量のファイル（16ファイル-16ディレクトリ、1ディレクトリあたり2GBファイル×1、`/dev/urandom` ブロックサイズ4KBの並列dd書き込み）

スクリプトを見つけるには、https://github.com/whyistheinternetbroken/NetAppFlexGroup/を参照してください。

テストの目的は、最大限のパフォーマンスを示すことではありません。そのためには、大規模なAFFシステム、クラスタ内のノード、クライアントなどを使用しました。

代わりに、これらのテストには次の3つの目標があります。

- に、さまざまなワークロードタイプに対するwsizeとrsizeの値の影響を示します。
- に、NFSv3、NFSv4.1、NFSv4.2のパフォーマンスの比較を示します。
- これらのワークロードに対するNFS Kerberosの効果を表示します。

これらのテストはそれぞれ5回実行され、より正確な結果を得るために値が平均化されました。

テスト環境に関する重要なポイントは次のとおりです。

- NFSv3、NFSv4.1、NFSv4.2（pNFSあり/なし）
- クライアント：
  - RHEL 8.3 VM×2
  - nconnect = 8
  - デフォルトのままのTCPスロットテーブル（65536）
  - 仮想メモリを40% `vm.dirty_ratio` および20%に調整 `vm.dirty_background_ratio`
- AFF A300 HAペア- 2ノード（ONTAP 9.9.1）
- FlexGroupボリューム（メンバーボリューム×16、ノードあたり8）
- 10Gbネットワーク

  注： このテストは、NFSの最大パフォーマンスや使用しているシステムを示すためのものではなく、同じネットワークとハードウェアでテストを実行してNFSのバージョンとブロックサイズを比較することを目的としています。

4KBのファイルテストの内訳でのNFS処理は次のようになります（作成/検索/書き込みも含む）。

```
Object: nfsv3
Instance: DEMO
    Counter                                                     Value
-------------------------------------------------------- --------------------------------------------------------
    access_total                                                 1017
    create_percent                                                 33%
    create_total                                               1000000
    fsinfo_total                                                     4
    getattr_total                                                    3
    lookup_percent                                                 33%
    lookup_total                                               1000003
    mkdir_total                                                   1003
    null_total                                                       4
    pathconf_total                                                   2
    total_ops                                                    36169
    write_percent                                                  33%
    write_total                                                1000000
```

2GBのファイルテストの内訳でのNFS処理は次のようになります（100%書き込み）。

```
Object: nfsv3
Instance: DEMO
Number of Constituents: 2 (complete_aggregation)
    Counter                                                     Value
    -------------------------------------------------------- --------------------------------------------------------
    access_total                                                    32
    create_total                                                    16
    fsinfo_total                                                     4
    getattr_total                                                   18
    lookup_total                                                    19
```

```
mkdir_total                                    19
null_total                                      4
pathconf_total                                  2
total_ops                                    3045
write_percent                                 99%
write_total                                511520
```

テスト1：ファイル数が多いテスト–多数のフォルダとファイル

このテストでは、1、000個のフォルダに約100万個の4KBファイルを作成し、マウントに複数のTCP転送サイズの値を使用しました。ファイル作成方法はPythonからのF.write呼び出しで、16文字を何度も何度も何度もファイルに書き込んで4KBに達する。これにより、NFS処理が増加しました。このテストでは、複数の異なる転送サイズ（NFSバージョン（NFSv3、NFSv4.1、NFSv4.2）とNFSv4.1（pNFSあり/なし）を使用し、nConnect=8を使用しました。

表35 に、各プロトコルの平均完了時間、平均IOPS、平均スループット、wsize/rsizeの値、CPUの平均ビジー率を示します。

`qos statistics performance show statistics show-periodic` この情報は、コマンド（レイテンシ、IOPS、スループット）とコマンド（CPUビジー率）を使用して収集されました。

表35）ファイル数の多いテスト結果（100万ファイル）

テスト	完了時間（秒）	平均 IOPS	平均MBps
NFSv3-64K wsize/rsize	～68.6	~55836	～72.6
NFSv3-256、000 wsize/rsize	～71	~55574	～72.2
NFSv3-1MB（wsize/rsize）	～73.1	~55865	～72.7
NFSv4.1-64K wsize/rsize	～251.1	～11182	～15.4
NFSv4.1-256K wsize/rsize	～259.5	～12041	～15.7
NFSv4.1-1MB wsize/rsize	～257.9	～11956年	～15.6
NFSv4.1-64K wsize/rsize（pNFS）	～254	～11818年	～15.4
NFSv4.1-256K wsize/rsize（pNFS）	～253.9	～11688	～15.2
NFSv4.1-1MB / rsize（pNFS）	～253.1	～11850年	～15.4
NFSv4.2-64K wsize/rsize（pNFS）	～256.3	～11756	～15.5
NFSv4.2-256K wsize/rsize（pNFS）	～255.5	～11890年	～15.4
NFSv4.2-1MB wsize/rsize（pNFS）	～256.1	～11764	～15.2

表36）ファイル数の多いテスト結果（100万個のファイル、CPUの平均ビジー率と平均レイテンシ）

テスト	平均合計レイテンシ（ミリ秒）	平均CPUビジー率
NFSv3-64K wsize/rsize	～15.4	最大64%
NFSv3-256、000 wsize/rsize	～15.1	約63%
NFSv3-1MB（wsize/rsize）	～15.3	約63%
NFSv4.1-64K wsize/rsize	～30.3	約27%
NFSv4.1-256K wsize/rsize	～29.8	約27%
NFSv4.1-1MB wsize/rsize	～30	約27%
NFSv4.1（pNFS）-64K wsize/rsize	～30.5	約26%
NFSv4.1（pNFS）-256K wsize/rsize	～30.9	約26%
NFSv4.1（pNFS）-1MB（wsize/rsize）	～30.4	約26%
NFSv4.2（pNFS）-64K wsize/rsize	～30.3	約26%

テスト	平均合計レイテンシ（ミリ秒）	平均CPUビジー率
NFSv4.2（pNFS）-256K wsize/rsize	～30.1	約26%
NFSv4.2（pNFS）-1MB wsize/rsize	～30.5	約27%

所見

- マウントのwsize / rsizeオプションに関係なく、ワークロードの性質（多数の小さなファイル）のためにほぼ同じ数のIOPSが生成されました。ブロックを64Kまたは1MBのチャンクで送信したかにかかわらず、それでも4Kファイルのみを送信したため、クライアントはそれらを1つずつ送信しました。NFSv3の平均最大スループットはあまり変化しませんでしたが、wsize / rsizeの値を大きくすると多少上昇しました。
- 全体的に見ると、ONTAP 9.9.1では、スループットとIOPSが高く、NFSv3はNFSv4.xよりもパフォーマンスが大幅に向上しました（最大3.5倍向上）。
- NFSv4.xでは全体のIOPSが少なくなるため、全体的なCPUビジー率が低くなりました。処理する処理数が少ない=使用するCPUが少なくなります。
- pNFSは、ここではあまりパフォーマンスを向上させませんでした。
- NFSv4.2では、NFSv4.1に比べてパフォーマンス上のメリットはありませんでした。

テスト2：ファイル数が少ないテスト–ファイル数が少ない、サイズが大きい

このテストでは、マルチスレッドdd処理を使用して2GBのファイル16個（フォルダ16個、フォルダ1個につきファイル1個）を作成しました。この処理では、各クライアントでフォルダごとのプロセスが作成されます。このテストでは、さまざまな転送サイズ、NFSバージョン（NFSv3、NFSv4.1、NFSv4.2）、NFSv4.x（pNFSあり/なし）を使用しました。ファイルの作成には、`dev/urandom` ストレージ効率化がファイルサイズにあまり影響しないように、/を使用して、4KBのブロックサイズとddを並行して使用しました。これにより、より純粋なスループットテストが生成されました。

表37）ファイル数の少ないテスト結果（2GBのファイル）

テスト	完了時間（秒）	平均 IOPS	平均MBps
NFSv3-64K wsize/rsize	～148.4	～3566	～222.8
NFSv3-256、000 wsize/rsize	～147.7	～901	～225.2
NFSv3-1MB（wsize/rsize）	～148.7	～226	～225
NFSv4.1-64K wsize/rsize	～148.6	～3578	～223.6
NFSv4.1-256K wsize/rsize	～147.6	～900	～224.8
NFSv4.1-1MB wsize/rsize	～148.1	～226	～225
NFSv4.1（pNFS）-64K wsize/rsize	～148.1	～3572	～222.4
NFSv4.1（pNFS）-256K wsize/rsize	～147.3	~902	～224.5
NFSv4.1（pNFS）-1MB（wsize/rsize）	～147.5	～228	～223.4
NFSv4.2（pNFS）-64K wsize/rsize	～147.9	～3593	～224.3
NFSv4.2（pNFS）-256K wsize/rsize	～148.1	～903	～224.6
NFSv4.2（pNFS）-1MB wsize/rsize	～147.7	～229	～224.3

表38）ファイル数の少ないテスト結果–CPUの平均ビジー率と平均レイテンシ

テスト	平均合計レイテンシ（ミリ秒）	平均CPUビジー率
NFSv3-64K wsize/rsize	～0.44	約28%
NFSv3-256、000 wsize/rsize	～0.87	約25%
NFSv3-1MB（wsize/rsize）	～3.98	約25%
NFSv4.1-64K wsize/rsize	～0.44	約30%

テスト	平均合計レイテンシ（ミリ秒）	平均CPUビジー率
NFSv4.1-256K wsize/rsize	～0.89	約27%
NFSv4.1-1MB wsize/rsize	～4.15	約26%
NFSv4.1（pNFS）-64K wsize/rsize	～0.31	約29%
NFSv4.1（pNFS）-256K wsize/rsize	～0.65	約25%
NFSv4.1（pNFS）-1MB（wsize/rsize）	～2.51	約26%
NFSv4.2（pNFS）-64K wsize/rsize	～0.35	約28%
NFSv4.2（pNFS）-256K wsize/rsize	～0.65	約25%
NFSv4.2（pNFS）-1MB wsize/rsize	～2.1	約25%

所見

- 一般的に、NFSv3とNFSv4.xのパフォーマンスはこのタイプのワークロードでほぼ同じです。テストの完了までの時間は1～2秒以内で、スループット値は全体的にほぼ同じでした。主な違いは平均レイテンシでした。
- NFSv4.xはNFSv3よりもレイテンシが低く、特にpNFSを使用してデータのローカリティを確保した場合に顕著でした。データ要求がクラスタネットワークを使用するようにすると、ワークロードへの書き込みレイテンシが0.1~1.5ミリ秒（mount wsizeに依存）、クラスタネットワーク全体で100~110MBのトラフィックが追加されます。
- NFSv4.2では、1MBのwsizeを使用した場合（約0.4ミリ秒短縮）を除いて、NFSv4.1からのパフォーマンスの大幅な向上は見られませんでした。
- ほとんどが読み取り/書き込みであるシーケンシャルワークロードの場合、pNFSを使用したNFSv4.1以降は、同じワークロードに対してNFSv3と同様に実行されます。

NFS Kerberosパフォーマンステスト

NFS Kerberosパフォーマンステストに関するこのセクションは削除されました。このドキュメントの今後のバージョンでは、修正された番号が公開される予定です。

ONTAPのデフォルトのNFSポート

表39）ONTAPのデフォルトのNFSポート

NFSサービス	ONTAPポート	該当するNFSバージョン	ポートを変更するオプション
mountd	635	NFSv3	-mountd-port
ポートマッパー	111	NFSv3	該当なし-変更不可
NLM	4045	NFSv3	-nlm-port
NSM	4046	NFSv3	-nsm-port
NFS	2049年	NFSv3 and NFSv4.x	N/A：変更不可
rquota	4049	NFSv3	-rquotad-port

Kelly Alexander、CPOC Labs、NetApp（Kerberosパフォーマンス情報）に感謝します。

詳細情報の入手方法

コメント要求（RFC）

- RFC 2203：『RPCSEC_GSS Protocol Specification』
- RFC 3530：『Network File System（NFS）Version 4 Protocol』
- RFC 5661：『Network File System（NFS）Version 4 Minor Version 1 Protocol』

- RFC 5331：『RPC：Remote手順Call Protocol Specification Version 2』

テクニカル レポート

- TR-3580：『NFSv4 Enhancements and Best Practices Guide：Data ONTAP Implementation』
- TR-3633：『NetApp ONTAPを基盤としたOracleデータベース』
- TR-4523：『ONTAPにおけるDNSロードバランシング』
- TR-4543：『SMB Best Practices、ONTAP 9.x』
- TR-4571：『NetApp FlexGroup Volume Best Practices and Implementation Guide』
- TR-4668：『Name Services Best Practices』
- TR-4616：『NFS Kerberos in ONTAP with Microsoft Active Directory』
- TR-4617：『Electronic Design Automation Best Practices』
- TR-4743：『FlexCache in ONTAP』
- TR-4835：『How to Configure LDAP in ONTAP』
- TR-4887：『Multiprotocol NAS in NetApp ONTAP−Overview and Best Practices』

バージョン履歴

バージョン	日付	ドキュメントの改訂履歴
バージョン1.0	2013年6月	初版リリース
バージョン2.0	2013年10月	ONTAP 8.2用に更新
バージョン2.1	2014年1月	ONTAP 8.2.1用に更新
バージョン2.2	2014年9月	ONTAP 8.2.2用に更新
バージョン3.0	2015年2月	ONTAP 8.3用に更新
バージョン3.1	2015年7月	ONTAP 8.3.1用に更新
バージョン3.2	2016年2月	ONTAP 8.3.2用に更新
バージョン4.0	2016年7月	ONTAP 9.0用に更新
バージョン 4.1	2016年10月	ONTAP 9.1用に更新
バージョン4.2	2017年7月	ONTAP 9.2用に更新
バージョン5.0	2020年6月	メジャーリビジョン；ONTAP 9.7用に更新
バージョン5.1	2021年2月	ONTAP 9.8向け更新版
バージョン5.2	2021年6月	ONTAP 9.9.1用に更新
バージョン5.3	2021年11月	ONTAP 9.10.1用に更新
バージョン5.3.1	2023年6月	Kerberosパフォーマンスの数値を削除

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-4067-0523-JP

NetApp