ホストの設定
Enterprise applications

NetApp
May 19, 2024

目次

ホストの設定

IBM AIXを使用するOracleデータベース

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

同時I/O

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

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

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

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

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

AIX NFSのマウントオプション

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

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

次の表に、RACのAIX NFSマウントオプションを示します。

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

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

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

AIX JFS / JFS2のマウントオプション

次の表に、AIX JFS / JFS2のマウントオプションを示します。

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

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

HP-UXを使用したOracleデータベース

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

HP-UX NFSのマウントオプション

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

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

次の表に、RACのHP-UX NFSマウントオプションを示します。

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

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

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

HP-UX VxFSマウントオプション

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

```
delaylog,nodatainlog
```

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

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

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

```
delaylog,cio
```

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

Linuxを使用したOracleデータベース

Linux OSに固有の設定に関するトピック。

Linux NFSv3 TCPスロットテーブル

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

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

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

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

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

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

Linux NFSのマウントオプション

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

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

次の表に、RACのLinux NFSマウントオプションを示します。

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

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

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

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

ACCESSとGETATTR

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

Linux Direct NFS

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

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

Linux Direct NFSとOracle RAC

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

するため、レイテンシが低下し、ストレージコントローラの負荷が増加します。

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

Linux Direct NFSと**oranfstab**ファイル

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

I/Oスケジューラ

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

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

マルチパス

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

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

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

これを修正するには、次のコマンドを使用します。

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

ASMミラーリング

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

ています。

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

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

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

```
polling_interval 5
no_path_retry 24
```

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

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

Linuxのxfs、ext3、ext4のマウントオプション

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

ASMLib/AFD（ASM Filter Driver）を使用するOracleデータベース

AFDとASMlibを使用するLinux OSに固有の設定トピック

ASMlibブロックサイズ

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

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

し、ASMデバイスの作成時にASMヘッダーを永続的にスタンプします。

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

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

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

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

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

ASMフィルタドライブ（AFD）のブロックサイズ

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

デバイスのブロックサイズ

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

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

と4KBのブロックサイズのASMディスクグループの間でファイルを複製できないことがあります。

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

マルチパステンソウサイス

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

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

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

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

Microsoft Windowsを使用したOracleデータベース

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

NFS

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

SAN

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

Solarisを使用したOracleデータベース

Solaris OSに固有の構成に関するトピック

Solaris NFSのマウントオプション

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

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

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

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

次の表に、Solaris NFSのマウントオプションを示します。

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

シングルインスタンスとRACマウントオプションの主な違いは、 `noac` および `forcedirectio` をマウント

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

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

Solaris UFSのマウントオプション

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

Solaris ZFS

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

mvector

Solaris 11では、大規模なI/O処理の処理方法が変更され、SANストレージアレイのパフォーマンスに重大な問題が発生する可能性がありました。この問題の詳細については、NetAppバグレポート630173「Solaris 11 ZFSパフォーマンスの回帰」を参照してください。"The解決策is to change a OS parameter called `zfs_mvector_max_size`。

rootとして次のコマンドを実行します。

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

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

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

カーネル

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

LUN構成

LUNを設定するには、次の手順を実行します。

1. タイプがのLUNを作成します。 `solaris`。
2. で指定された適切なHost Utility Kit（HUK）をインストールします。 "ネットアップの Interoperability Matrix Tool （ IMT ）"。

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

zpools

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

の値 `ashift` デフォルトは9です。これは2^9、つまり512バイトを意味します。最適なパフォーマンスを実現するには、 `ashift` 値は12（2^12=4K）である必要があります。この値はzpoolの作成時に設定され、変更することはできません。つまり、 `ashift` 12以外の場合は、新しく作成したzpoolにデータをコピーして移行する必要があります。

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

zpoolとSolaris LDOM

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

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

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

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

2. ZFSプールに使用するデバイスのVDCインスタンスを取得します。

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

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

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

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

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

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

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

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

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

ZFSインテントログ（**ZIL**）

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

ロバイアス

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

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

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

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

ダイレクト**I/O**

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

複数の**zpool**

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

インに戻してデータベースをリカバリできます。

ファイルシステムオプション

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