



Microsoft SQL Server の場合

Enterprise applications

NetApp
May 03, 2024

目次

| | |
|--|----|
| Microsoft SQL Server の場合 | 1 |
| ONTAP上のMicrosoft SQL Server | 1 |
| データベース設定 | 2 |
| ストレージ構成 | 9 |
| NetApp管理ソフトウェアによるMicrosoft SQL Serverデータ保護 | 23 |
| ONTAPを使用したMicrosoft SQL Serverディザスタリカバリ | 24 |
| ONTAP上のMicrosoft SQL Serverのセキュリティ保護 | 24 |

Microsoft SQL Server の場合

ONTAP上のMicrosoft SQL Server

ONTAPは解決策、Microsoft SQL Serverデータベースにエンタープライズクラスのセキュリティとパフォーマンスを提供すると同時に、環境を管理するためのワールドクラスのツールも提供します。



このドキュメントは、以前に公開されたテクニカルレポート TR-4590 : 『Best Practice Guide for Microsoft SQL Server with ONTAP』 の内容を置き換えます。

NetAppは、読者が以下について実用的な知識を持っていることを前提としています。

- ONTAP ソフトウェア
- バックアップソフトウェアとしてのNetApp SnapCenterには、次のものが含まれます。
 - SnapCenter Plug-in for Microsoft Windows の略
 - SQL Server向けSnapCenterプラグイン
- Microsoft SQL Serverのアーキテクチャと管理

このベストプラクティスセクションの範囲は、NetAppがストレージインフラに推奨する設計原則と推奨される標準に基づいた技術設計に限定されます。エンドツーエンドの実装は範囲外です。

NetApp製品間での設定の互換性については、[を参照してください。 "ネットアップの Interoperability Matrix Tool \(IMT\) "](#)。

Microsoft SQL Serverのワークロード

SQL Serverを導入する前に、SQL Serverインスタンスがサポートするアプリケーションのデータベースワークロード要件を理解しておく必要があります。容量、パフォーマンス、可用性に関する要件はアプリケーションごとに異なるため、各データベースはこれらの要件を最適にサポートするように設計する必要があります。多くの組織では、アプリケーション要件を使用してSLAを定義し、データベースを複数の管理階層に分類しています。SQL Serverのワークロードは次のように記述できます。

- OLTPデータベースは、多くの場合、組織で最も重要なデータベースでもあります。これらのデータベースは通常、顧客向けのアプリケーションをバックアップし、企業の中核業務に不可欠であると考えられています。ミッションクリティカルなOLTPデータベースとサポート対象のアプリケーションには、高レベルのパフォーマンスが必要で、パフォーマンスの低下や可用性の影響を受けやすいSLAが設定されていることがよくあります。また、Always-OnフェイルオーバークラスターやAlways-On可用性グループの候補になることもあります。これらのタイプのデータベースのI/O構成は、通常、ランダムリードが75⁹⁰%、書き込みが25¹⁰%という特徴があります。
- 意思決定支援システム(DSS)データベースは、データウェアハウスとも呼ばれます。これらのデータベースは、ビジネスの分析に依存している多くの組織でミッションクリティカルです。これらのデータベースは、クエリの実行時にCPU利用率やディスクからの読み取り処理の影響を受けます。多くの組織では、DSSデータベースは月、四半期、年末に最も重要なデータベースです。このワークロードは、一般に100%の読み取りI/O構成です。

データベース設定

Microsoft SQL ServerのCPU構成

システムパフォーマンスを向上させるには、SQL Serverの設定とサーバ構成を変更して、適切な数のプロセッサを実行に使用する必要があります。

ハイパースレッディング

ハイパースレッディングはインテル独自の同時マルチスレッド(SMT)実装であり、x86マイクロプロセッサ上で実行される計算（マルチタスク）の並列化を改善します。

ハイパースレッディングを使用するハードウェアでは、論理ハイパースレッディングCPUを物理CPUとしてオペレーティングシステムに認識させることができます。SQL Serverは、オペレーティングシステムが提供する物理CPUを認識し、ハイパースレッドプロセッサを使用できます。これにより、並列化が促進され、パフォーマンスが向上します。

ここで注意すべき点は、SQL Serverの各バージョンには、使用できるコンピューティング能力に独自の制限があることです。詳細については、「SQL Serverのエディション別のコンピューティング容量制限」を参照してください。

SQL Serverのライセンスには2つのオプションがあります。1つ目はサーバ+クライアントアクセスライセンス（CAL）モデルと呼ばれ、2つ目はプロセッサごとのコアモデルです。SQL Serverで利用可能なすべての製品機能には、サーバ+ CAL戦略でアクセスできますが、ソケットあたりのCPUコア数はハードウェアで20に制限されています。ソケットあたり20個以上のCPUコアを搭載したサーバ用のSQL Server Enterprise Edition+CALがある場合でも、アプリケーションはそのインスタンスですべてのコアを一度に使用することはできません。

次の図は、起動後のSQL Serverログメッセージを示しています。これは、コア制限の適用を示しています。

ログエントリは、**SQL Server**の起動後に使用されているコアの数を示します。

```

2017-01-11 07:16:30.71 Server      Microsoft SQL Server 2016
(RTM) - 13.0.1601.5 (X64)
Apr 29 2016 23:23:58
Copyright (c) Microsoft Corporation
Enterprise Edition (64-bit) on Windows Server 2016
Datacenter 6.3 <X64> (Build 14393: )

2017-01-11 07:16:30.71 Server      UTC adjustment: -8:00
2017-01-11 07:16:30.71 Server      (c) Microsoft Corporation.
2017-01-11 07:16:30.71 Server      All rights reserved.
2017-01-11 07:16:30.71 Server      Server process ID is 10176.
2017-01-11 07:16:30.71 Server      System Manufacturer:
'FUJITSU', System Model: 'PRIMERGY RX2540 M1'.
2017-01-11 07:16:30.71 Server      Authentication mode is MIXED.
2017-01-11 07:16:30.71 Server      Logging SQL Server messages
in file 'C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\Log\ERRORLOG'.
2017-01-11 07:16:30.71 Server      The service account is 'SEA-
TM\FUJIA2R30$'. This is an informational message; no user action
is required.
2017-01-11 07:16:30.71 Server      Registry startup parameters:
-d C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\DATA\master.mdf
-e C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\Log\ERRORLOG
-l C:\Program Files\Microsoft SQL Server
\MSSQL13.MSSQLSERVER\MSSQL\DATA\mastlog.ldf
-T 3502
-T 834
2017-01-11 07:16:30.71 Server      Command Line Startup
Parameters:
-a "MSSQLSERVER"
2017-01-11 07:16:30.72 Server      SQL Server detected 2 sockets
with 18 cores per socket and 36 logical processors per socket,
72 total logical processors; using 40 logical processors based
on SQL Server licensing. This is an informational message; no
user action is required.
2017-01-11 07:16:30.72 Server      SQL Server is starting at

```

したがって、すべてのCPUを使用するには、プロセッサ単位のコアライセンスを使用する必要があります。SQL Serverのライセンスの詳細については、[を参照してください。"SQL Server 2022：最新データプラットフォームを実現"](#)。

CPUアフィニティ

パフォーマンスの問題が発生しない限り、プロセッサアフィニティのデフォルトを変更する必要はありませんが、その内容と動作を理解する価値はあります。

SQL Serverは、次の2つのオプションでプロセッサアフィニティをサポートします。

- CPUアフィニティマスク
- アフィニティI/Oマスク

SQL Serverは、オペレーティングシステムで使用可能なすべてのCPUを使用します（プロセッサ単位のコアライセンスが選択されている場合）。すべてのCPUにスケジューラを作成し、特定のワークロードでリソースを最大限に活用します。マルチタスクを実行する場合、オペレーティングシステムやサーバー上のその他のアプリケーションは、プロセススレッドをプロセッサ間で切り替えることができます。SQL Serverはリソースを大量に消費するアプリケーションであるため、この状況が発生するとパフォーマンスに影響する可能性があります。影響を最小限に抑えるには、SQL Serverのすべての負荷が事前に選択されたプロセッサグループに送られるようにプロセッサを構成します。これは、CPUアフィニティマスクを使用することによって実現されます。

アフィニティI/Oマスクオプションは、SQL ServerディスクI/OをCPUのサブセットにバインドします。SQL Server OLTP環境では、この拡張により、I/O処理を実行するSQL Serverスレッドのパフォーマンスが向上します。

並列処理の最大回数(MAXDOP)

デフォルトでは、プロセッサ単位のコアライセンスが選択されている場合、SQL Serverはクエリの実行中に使用可能なすべてのCPUを使用します。

これは大規模なクエリには役立ちますが、原因のパフォーマンスが低下し、同時実行数が制限される可能性があります。1つのCPUソケット内の物理コアの数に並列処理を制限する方法が適しています。たとえば、ソケットあたり12コアの2つの物理CPUソケットを持つサーバでは、ハイパースレディングに関係なく、MAXDOPを12に設定する必要があります。MAXDOPでは、使用するCPUを制限したり、指定したりすることはできません。代わりに、単一のバッチクエリで使用できるCPUの数を制限します。



* NetAppでは、データウェアハウスなどのDSSでは、MAXDOP 50から開始し、必要に応じてチューニングアップまたはチューニングダウンを検討することを推奨しています。変更を加えるときは、必ずアプリケーション内の重要なクエリを測定してください。

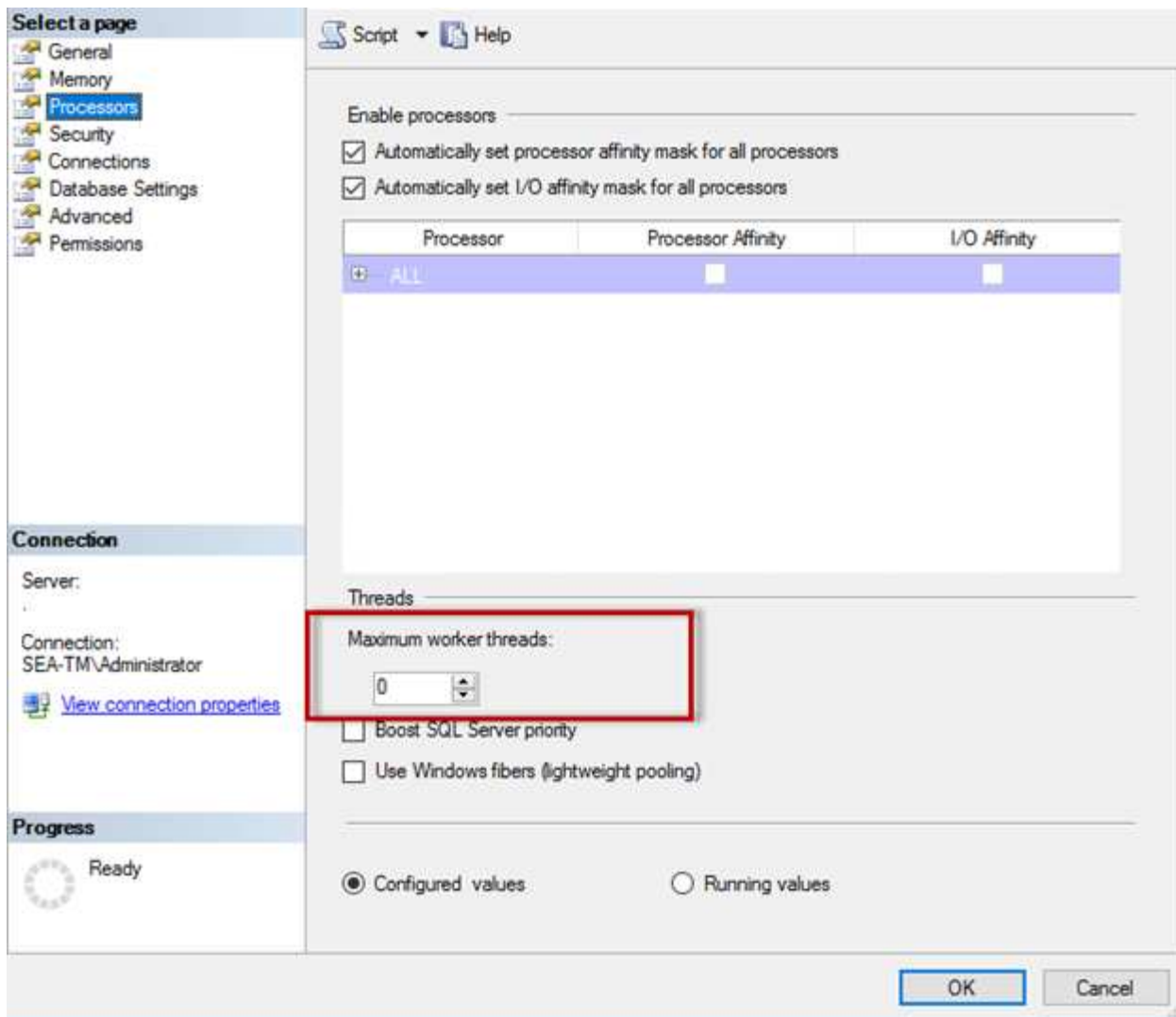
ワーカースレッドの最大数

最大ワーカースレッド数オプションは、多数のクライアントがSQL Serverに接続されている場合にパフォーマンスを最適化するのに役立ちます。

通常、クエリ要求ごとに個別のオペレーティングシステムスレッドが作成されます。SQL Serverへの同時接続が数百もの場合、クエリ要求ごとに1つのスレッドが大量のシステムリソースを消費します。最大ワーカースレッド数オプションを使用すると、SQL Serverでワーカースレッドのプールを作成して多数のクエリ要求を処理できるようになるため、パフォーマンスが向上します。

デフォルト値は0で、SQL Serverは起動時にワーカースレッド数を自動的に設定できます。これはほとんどのシステムで機能します。ワーカースレッドの最大数は高度なオプションであり、経験豊富なデータベース管理者（DBA）の支援なしに変更しないでください。

より多くのワーカースレッドを使用するようにSQL Serverを設定する必要があるのはいつですか？各スケジューラの平均ワークキューの長さが1を超える場合は、負荷がCPUに制限されていないか、その他の重い待機時間が発生している場合にのみ、システムにスレッドを追加することでメリットが得られます。これらのいずれかが発生している場合、スレッドを追加しても、他のシステムボトルネックを待つことになるため、効果はありません。最大ワーカースレッド数の詳細については、["max worker threadsサーバ設定オプションの設定"](#)を参照してください。



SQL Server Management Studioを使用した最大ワーカースレッド数の設定。

The following example shows how to configure the max work threads option using T-SQL.

```
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE ;
GO
EXEC sp_configure 'max worker threads', 900 ;
GO
RECONFIGURE;
GO
```

Microsoft SQL Serverのメモリ構成

次のセクションでは、データベースのパフォーマンスを最適化するためのSQL Serverメ

メモリ設定の構成について説明します。

最大サーバメモリ

max server memoryオプションは、SQL Serverインスタンスで使用できるメモリの最大容量を設定します。

通常、SQL Serverが実行されている同じサーバで複数のアプリケーションが実行されていて、これらのアプリケーションが正常に機能するのに十分なメモリを確保したい場合に使用されます。

アプリケーションによっては、起動時に使用可能なメモリのみを使用し、必要に応じて要求しないものもあります。ここで、最大サーバメモリ設定が有効になります。

複数のSQL Serverインスタンスを持つSQL Serverクラスタでは、各インスタンスがリソースを競合する可能性があります。SQL Serverインスタンスごとにメモリ制限を設定すると、各インスタンスのパフォーマンスを最大限に高めることができます。



* NetAppでは、パフォーマンスの問題を回避するために、オペレーティングシステム用に少なくとも4GBから6GBのRAMを残しておくことを推奨しています。

Select a page

- General
- Memory
- Processors
- Security
- Connections
- Database Settings
- Advanced
- Permissions

Script Help

Server memory options

Minimum server memory (in MB):
0

Maximum server memory (in MB):
120832

Other memory options

Index creation memory (in KB, 0 = dynamic memory):
0

Minimum memory per query (in KB):
1024

Connection

Server:
SEA-TM\Administrator

View connection properties

Progress

Ready

Configured values Running values

OK Cancel

SQL Server Management Studioを使用したサーバの最小メモリと最大メモリの調整

SQL Server Management Studioを使用してサーバの最小メモリまたは最大メモリを調整するには、SQL Serverサービスを再起動する必要があります。次のコードを使用して、Transact SQL (T-SQL) を使用してサーバメモリを調整できます。

```
EXECUTE sp_configure 'show advanced options', 1
GO
EXECUTE sp_configure 'min server memory (MB)', 2048
GO
EXEC sp_configure 'max server memory (MB)', 120832
GO
RECONFIGURE WITH OVERRIDE
```

不均一なメモリアクセス

NUMA (Nonuniform Memory Access) は、プロセッサバスの負荷を増やすことなくプロセッサ速度を向上させるメモリアクセス最適化方法です。

SQL ServerがインストールされているサーバでNUMAが構成されている場合、SQL ServerはNUMAを認識し、NUMAハードウェアで優れたパフォーマンスを発揮するため、追加の構成は必要ありません。

インデックス作成メモリ

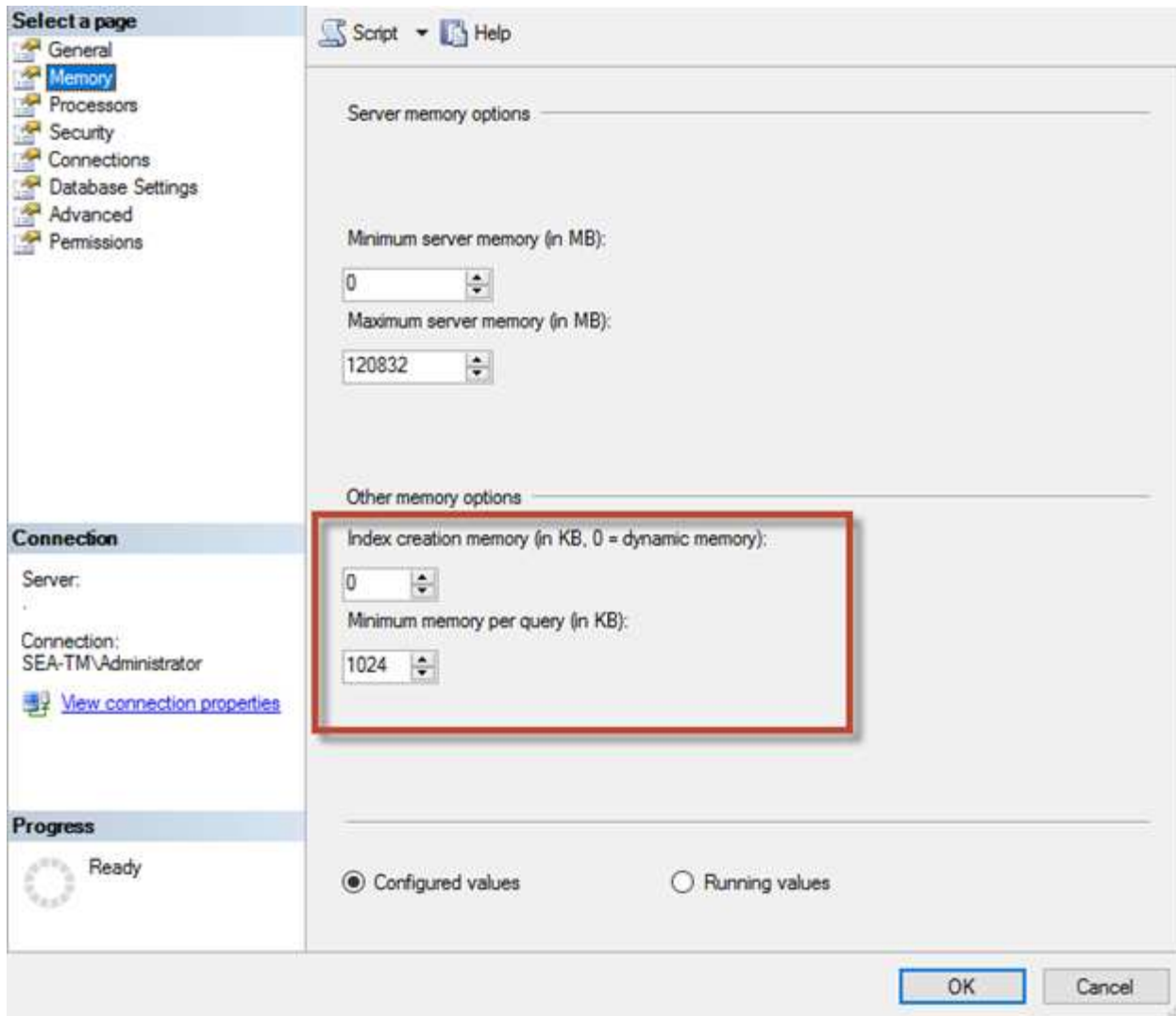
index create memoryオプションは、通常は変更しないもう1つの高度なオプションです。

インデックスを作成するために最初に割り当てられるRAMの最大容量を制御します。このオプションのデフォルト値は0です。これは、SQL Serverによって自動的に管理されることを意味します。ただし、インデックスの作成に問題がある場合は、このオプションの値を増やすことを検討してください。

クエリあたりの最小メモリ

クエリを実行すると、SQL Serverは効率的に実行するために最適なメモリ容量を割り当てようとします。

デフォルトでは、クエリごとの最小メモリ設定では、実行するクエリごとに>=が1024KBに割り当てられません。SQL Serverがインデックス作成処理に割り当てられるメモリ容量を動的に管理できるようにするには、この設定をデフォルト値の0のままにしておくことを推奨します。ただし、SQL ServerのRAM容量が効率的に実行するために必要な容量を超えている場合は、この設定を大きくすると、一部のクエリのパフォーマンスが向上することがあります。したがって、SQL Server、その他のアプリケーション、またはオペレーティングシステムで使用されていないサーバ上のメモリが使用可能であれば、この設定を大きくすることでSQL Server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。空きメモリがない場合、この設定を増やすと、全体的なパフォーマンスが低下する可能性があります。



バッファプール拡張

バッファプール拡張機能を使用すると、NVRAM拡張機能とデータベースエンジンのバッファプールをシームレスに統合して、I/Oスループットを大幅に向上させることができます。

バッファプール拡張機能は、すべてのSQL Serverエディションで使用できるわけではありません。64ビットのSQL Server Standard、Business Intelligence、およびEnterpriseエディションでのみ使用できます。

バッファプール拡張機能は、不揮発性ストレージ（通常はSSD）を使用してバッファプールキャッシュを拡張します。この拡張機能により、バッファプールはより大規模なデータベースワーキングセットに対応できるようになり、RAMとSSD間のI/Oのページングが強制され、小さなランダムI/OがメカニカルディスクからSSDに効果的にオフロードされます。SSDのレイテンシが低く、ランダムI/Oのパフォーマンスが向上するため、バッファプールを拡張することでI/Oスループットが大幅に向上します。

バッファプール拡張機能には、次の利点があります。

- ランダムI/Oスループットの向上
- I/Oレイテンシの低減
- トランザクションスループットの向上
- ハイブリッドバッファプールの拡張による読み取りパフォーマンスの向上

- 既存および将来の低コストメモリを活用できるキャッシュアーキテクチャ



- NetAppでは、バッファプール拡張を次のように設定することを推奨しています。
- バッファプール拡張ターゲットディスクとして使用できるように、SSDベースのLUN（NetApp AFFなど）がSQL Serverホストに提供されていることを確認します。
- 拡張ファイルのサイズは、バッファプールと同じかそれよりも大きくする必要があります。

次に、バッファプール拡張を32GBに設定するT-SQLコマンドの例を示します。

```
USE master
GO
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'P:\BUFFER POOL EXTENSION\SQLServerCache.BUFFER POOL
EXTENSION', SIZE = 32 GB);
GO
```

Microsoft SQL Server共有インスタンスと専用インスタンス

複数のSQL Serverは、サーバごとに1つのインスタンスとして構成することも、複数のインスタンスとして構成することもできます。通常、適切な決定は、サーバを本番用と開発用のどちらに使用するか、インスタンスがビジネスの運用やパフォーマンスの目標にとって重要であると判断されるかなどの要因によって決まります。

共有インスタンスの構成は、最初は簡単に設定できますが、リソースが分割されたりロックされたりする問題が発生し、共有SQL Serverインスタンスでデータベースがホストされている他のアプリケーションでパフォーマンスの問題が発生する可能性があります。

どのインスタンスがルート原因であるかを把握する必要があるため、パフォーマンスの問題のトラブルシューティングは複雑になります。この質問は、オペレーティングシステムライセンスとSQL Serverライセンスのコストと比較して検討されます。アプリケーションのパフォーマンスを最優先する場合は、専用インスタンスを使用することを推奨します。

Microsoftでは、SQL Serverのライセンスは、インスタンス単位ではなく、コア単位でサーバレベルで付与されます。このため、データベース管理者は、ライセンスコストを削減するために、サーバで処理できる数のSQL Serverインスタンスをインストールしようとします。これは、後で大きなパフォーマンスの問題につながる可能性があります。



- * NetAppでは、最適なパフォーマンスを得るために、可能な限り専用のSQL Serverインスタンスを選択することを推奨しています。

ストレージ構成

Microsoft SQL Serverストレージに関する考慮事項

ONTAPストレージソリューションとMicrosoft SQL Serverを組み合わせることで、今日の最も要求の厳しいアプリケーション要件を満たすエンタープライズレベルのデータベースストレージ設計を作成できます。

両方のテクノロジーを最適化するには、SQL ServerのI/Oパターンと特性を理解することが重要です。SQL Serverデータベース用の適切に設計されたストレージレイアウトは、SQL ServerのパフォーマンスとSQL Serverインフラの管理をサポートします。また、ストレージレイアウトを適切に配置すれば、初期導入を成功させ、ビジネスの成長に合わせて環境をスムーズに拡張できます。

データストレージ設計

SnapCenter を使用してバックアップを実行しないSQL Serverデータベースについては、データファイルとログファイルを別々のドライブに配置することを推奨します。データを同時に更新して要求するアプリケーションでは、ログファイルに書き込み負荷がかかり、（アプリケーションによっては）データファイルの読み取り/書き込み負荷が高くなります。データを取得する場合、ログファイルは必要ありません。そのため、データの要求は、そのドライブに配置されたデータファイルから満たすことができます。

新しいデータベースを作成するときは、データとログ用に別々のドライブを指定することを推奨します。データベース作成後にファイルを移動するには、データベースをオフラインにする必要があります。Microsoftのその他の推奨事項については、"[データファイルとログファイルを別々のドライブに配置](#)"。

アグリゲート

アグリゲートは、NetAppストレージ構成の最下位レベルのストレージコンテナです。一部のレガシードキュメントはインターネット上に存在し、異なるドライブセットにIOを分離することを推奨しています。これはONTAPでは推奨されません。NetAppは、データファイルとトランザクションログファイルを分離した共有アグリゲートと専用アグリゲートを使用して、さまざまなI/Oワークロードの特性評価テストを実施してきました。このテストでは、1つの大規模アグリゲートに複数のRAIDグループとドライブを配置することで、ストレージのパフォーマンスが最適化され、向上し、管理者が管理しやすくなることがわかりました。その理由は次の2つです。

- 1つの大きなアグリゲートで、すべてのドライブのI/O機能をすべてのファイルで使用できます。
- 1つの大きなアグリゲートで、最も効率的なディスクスペースを使用できます。

高可用性（HA）を実現するには、SQL Server Always On可用性グループのセカンダリ同期レプリカを、アグリゲート内の別のStorage Virtual Machine（SVM）に配置します。ディザスタリカバリを目的とした場合は、DRサイト内の別のストレージクラスタの一部であるアグリゲートに非同期レプリカを配置し、NetApp SnapMirrorテクノロジーを使用してコンテンツをレプリケートします。NetAppでは、ストレージのパフォーマンスを最適化するために、アグリゲートに利用可能な空きスペースを少なくとも10%確保することを推奨しています。

個のボリューム

NetApp FlexVolボリュームはアグリゲート内に作成され、格納されます。ONTAPボリュームがLUNではないため、この用語を使用すると混乱が生じることがあります。ONTAPボリュームはデータの管理コンテナです。ボリュームには、ファイル、LUN、さらにはS3オブジェクトが含まれている可能性があります。ボリュームはスペースを消費せず、格納されたデータの管理にのみ使用されます。

データベースボリュームの設計を作成する前に、SQL ServerのI/Oパターンと特性がワークロードやバックアップとリカバリの要件に応じてどのように変わるかを理解しておくことが重要です。フレキシブルボリュームについては、NetAppに関する次の推奨事項を参照してください。

- ホスト間でのボリュームの共有は避けてください。たとえば、1つのボリュームに2つのLUNを作成し、各LUNを別のホストで共有することは可能ですが、管理が複雑になる可能性があるため、この方法は避けてください。
- ドライブレターではなくNTFSマウントポイントを使用して、Windowsのドライブレターの制限（26文字）を超えます。ボリュームマウントポイントを使用する場合は、ボリュームラベルにマウントポイントと同じ名前を付けることを一般的に推奨します。
- 必要に応じて、スペース不足が発生しないようにボリュームのオートサイズポリシーを設定します。17 ONTAPを使用したMicrosoft SQL Serverのベストプラクティスガイド©2022 NetApp, Inc. 無断転載を禁じます。
- SQL ServerをSMB共有にインストールする場合は、フォルダを作成するためにSMB/CIFSボリュームでUnicodeが有効になっていることを確認してください。
- 運用面からの監視を容易にするために、ボリュームのスナップショット予約値をゼロに設定します。
- Snapshotスケジュールと保持ポリシーを無効にします。代わりに、SnapCenterを使用してSQL ServerデータボリュームのSnapshotコピーを調整します。
- SQL Serverシステムデータベースを専用ボリュームに配置します。
- tempdbは、特にI/O負荷の高いDBCC CHECKDB処理のために、SQL Serverが一時的なワークスペースとして使用するシステムデータベースです。したがって、このデータベースは、独立したスピンドルセットを持つ専用ボリュームに配置します。ボリューム数が課題となる大規模な環境では、慎重に計画を立てたあと、tempdbを少数のボリュームに統合し、他のシステムデータベースと同じボリュームに格納できます。tempdbのデータ保護は、SQL Serverを再起動するたびにこのデータベースが再作成されるため、優先度の高いものではありません。
- ランダムな読み取り/書き込みワークロードであるため、ユーザデータファイル (.mdf) を別々のボリュームに配置します。トランザクションログバックアップは、データベースバックアップよりも頻繁に作成するのが一般的です。そのため、トランザクションログファイル (.ldf) をデータファイルとは別のボリュームまたはVMDKに配置して、それぞれに個別のバックアップスケジュールを作成できるようにします。この分離により、ログファイルのシーケンシャルライトI/Oがデータファイルのランダムリード/ライトI/Oから分離され、SQL Serverのパフォーマンスが大幅に向上します。

LUN

- ユーザデータベースファイルとログバックアップを格納するログディレクトリが別々のボリュームにあることを確認して、SnapVaultテクノロジーでSnapshotが使用されている場合に保持ポリシーによって上書きされないようにしてください。
- SQL Serverデータベースが、フルテキスト検索関連ファイルなど、データベース以外のファイルを持つLUNとは別のLUN上に存在することを確認します。
- データベースのセカンダリファイルを（ファイルグループの一部として）別々のボリュームに配置すると、SQL Serverデータベースのパフォーマンスが向上します。この分離は、データベースの.mdfファイルがLUNを他の.mdfファイルと共有していない場合にのみ有効です。
- DiskManagerなどのツールを使用してLUNを作成する場合は、LUNをフォーマットするときに、パーティションの割り当て単位サイズが64Kに設定されていることを確認してください。
- を参照してください ["最新SANに対するONTAPのベストプラクティスに基づくMicrosoft Windowsとネイ](#)

タイプMPIO" WindowsのマルチパスサポートをMPIOプロパティのiSCSIデバイスに適用するには、次の手順を実行します。

Microsoft SQL Serverデータベースファイルおよびファイルグループ

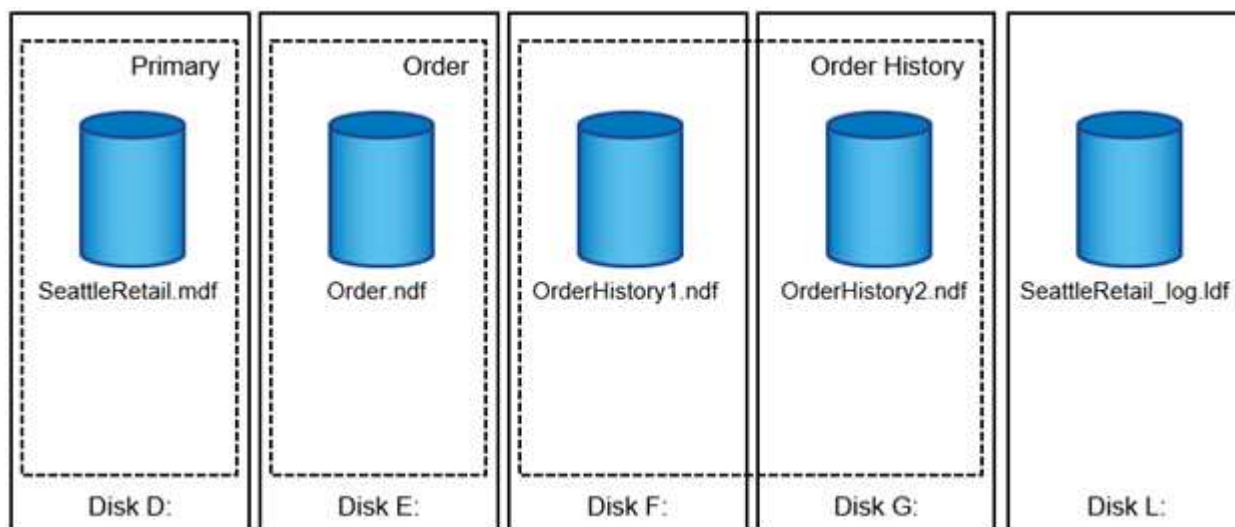
初期導入段階では、SQL ServerデータベースファイルをONTAPに適切に配置することが重要です。これにより、パフォーマンス、スペース管理、バックアップとリストアの最適な時間が確保され、ビジネス要件に合わせて設定できます。

理論的には、SQL Server (64ビット) ではインスタンスあたり32、767個のデータベースと524、272TBのデータベースサイズがサポートされますが、通常のインストールでは複数のデータベースが使用されます。ただし、SQL Serverで処理できるデータベースの数は、負荷とハードウェアによって異なります。SQL Serverインスタンスでは、数十、数百、場合によっては数千の小規模データベースをホストしていることも珍しくありません。

各データベースは、1つ以上のデータファイルと1つ以上のトランザクションログファイルで構成されます。トランザクションログには、データベーストランザクションに関する情報と、各セッションで行われたすべてのデータ変更が格納されます。データが変更されるたびに、SQL Serverはトランザクションログに十分な情報を格納して、アクションを元に戻す（ロールバックする）か、やり直す（再生する）かを指定します。SQL Serverトランザクションログは、データの整合性と堅牢性に関するSQL Serverの評価に不可欠な要素です。トランザクションログは、SQL Serverの不可分性、整合性、分離、耐久性（ACID）機能に不可欠です。SQL Serverは、データページが変更されるとすぐにトランザクションログに書き込みます。すべてのData Manipulation Language (DML) ステートメント（SELECT、INSERT、UPDATE、DELETEなど）は完全なトランザクションであり、トランザクションログによってセットベースの操作全体が確実に実行され、トランザクションの不可分性が保証されます。

各データベースには1つのプライマリデータファイルがあり、デフォルトでは.mdf拡張子が付いています。また、各データベースにセカンダリデータベースファイルを含めることもできます。これらのファイルには、デフォルトで.ndf拡張子が付いています。

すべてのデータベースファイルはファイルグループにグループ化されます。ファイルグループは論理ユニットであり、データベース管理を簡素化します。論理オブジェクトの配置と物理データベースファイルを分離できます。データベースオブジェクトテーブルを作成するときは、基になるデータファイルの設定を気にすることなく、配置するファイルグループを指定します。



ファイルグループ内に複数のデータファイルを配置できるため、複数のストレージデバイスに負荷を分散し

て、システムのI/Oパフォーマンスを向上させることができます。一方、SQL Serverはトランザクションログにシーケンシャルに書き込むため、トランザクションログには複数のファイルを使用するメリットはありません。

ファイルグループ内の論理オブジェクトの配置と物理データベースファイルの配置を分離することで、データベースファイルのレイアウトを微調整し、ストレージサブシステムを最大限に活用できます。たとえば、異なる顧客に自社製品を導入している独立系ソフトウェアベンダー（ISV）は、基盤となるI/O構成と導入段階で予想されるデータ量に基づいてデータベースファイルの数を調整できます。これらの変更は、データベースファイルではなくファイルグループにデータベースオブジェクトを配置するアプリケーション開発者には透過的です。



* NetAppでは、システムオブジェクト以外にプライマリファイルグループを使用しないことを推奨しています。ユーザオブジェクト用に別のファイルグループまたはファイルグループのセットを作成すると、特に大規模なデータベースの場合、データベースの管理とディザスタリカバリが容易になります。

データベースを作成するとき、または既存のデータベースに新しいファイルを追加するとき、初期ファイルサイズと自動拡張パラメータを指定できます。SQL Serverでは、Proportional Fill Algorithmを使用して、データを書き込むデータファイルを選択します。ファイルで使用可能な空きスペースに比例してデータ量が書き込まれます。ファイル内の空きスペースが多いほど、処理する書き込み数も多くなります。



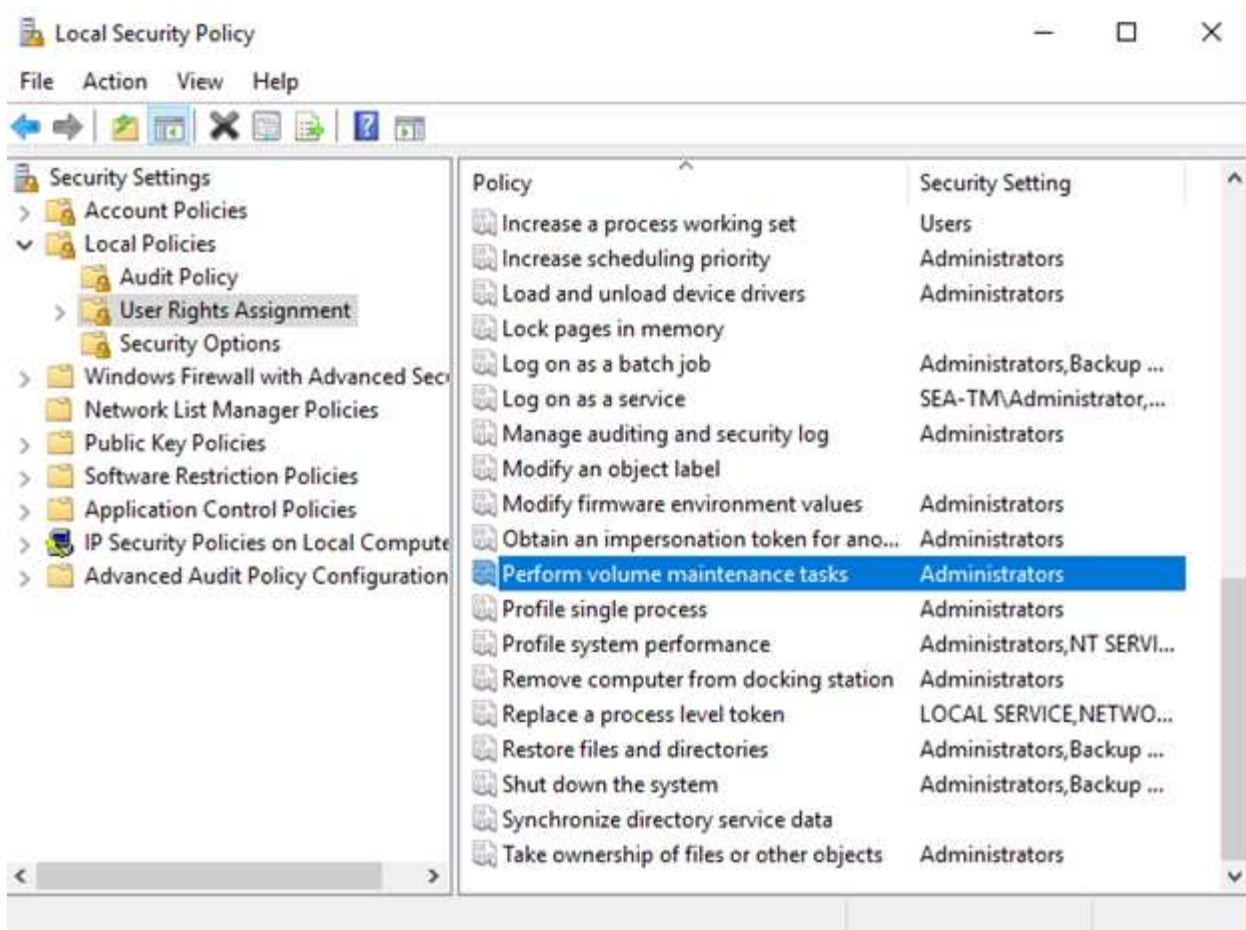
* NetAppでは、1つのファイルグループ内のすべてのファイルに同じ初期サイズと自動拡張パラメータを設定し、拡張サイズをパーセンテージではなくメガバイト単位で定義することを推奨しています*。これにより、Proportional Fill Algorithmは、データファイル間で書き込みアクティビティのバランスを均等に調整できます。

SQL Serverは、ファイルを拡張するたびに、新しく割り当てられたスペースをゼロでいっぱいにします。このプロセスは、対応するファイルへの書き込みが必要なすべてのセッションをブロックします。トランザクションログが増加した場合は、トランザクションログレコードを生成します。

SQL Serverは常にトランザクションログをゼロにし、その動作を変更することはできません。ただし、インスタントファイルの初期化を有効または無効にすることで、データファイルを初期化するかどうかを制御できます。インスタントファイルの初期化を有効にすると、データファイルの増加を高速化し、データベースの作成やリストアに必要な時間を短縮できます。

インスタントファイルの初期化には、わずかなセキュリティリスクが伴います。このオプションを有効にすると、データファイルの未割り当て部分に、以前に削除されたOSファイルの情報を含めることができます。データベース管理者はこのようなデータを調べることができます。

インスタントファイルの初期化を有効にするには、「ボリュームメンテナンスタスクの実行」とも呼ばれるSA_MANAGE_VOLUME_name権限をSQL Serverスタートアップアカウントに追加します。これは、次の図に示すように、ローカルセキュリティポリシー管理アプリケーション（secpol.msc）で実行できます。「Perform volume maintenance task」権限のプロパティを開き、SQL Serverスタートアップアカウントをユーザのリストに追加します。



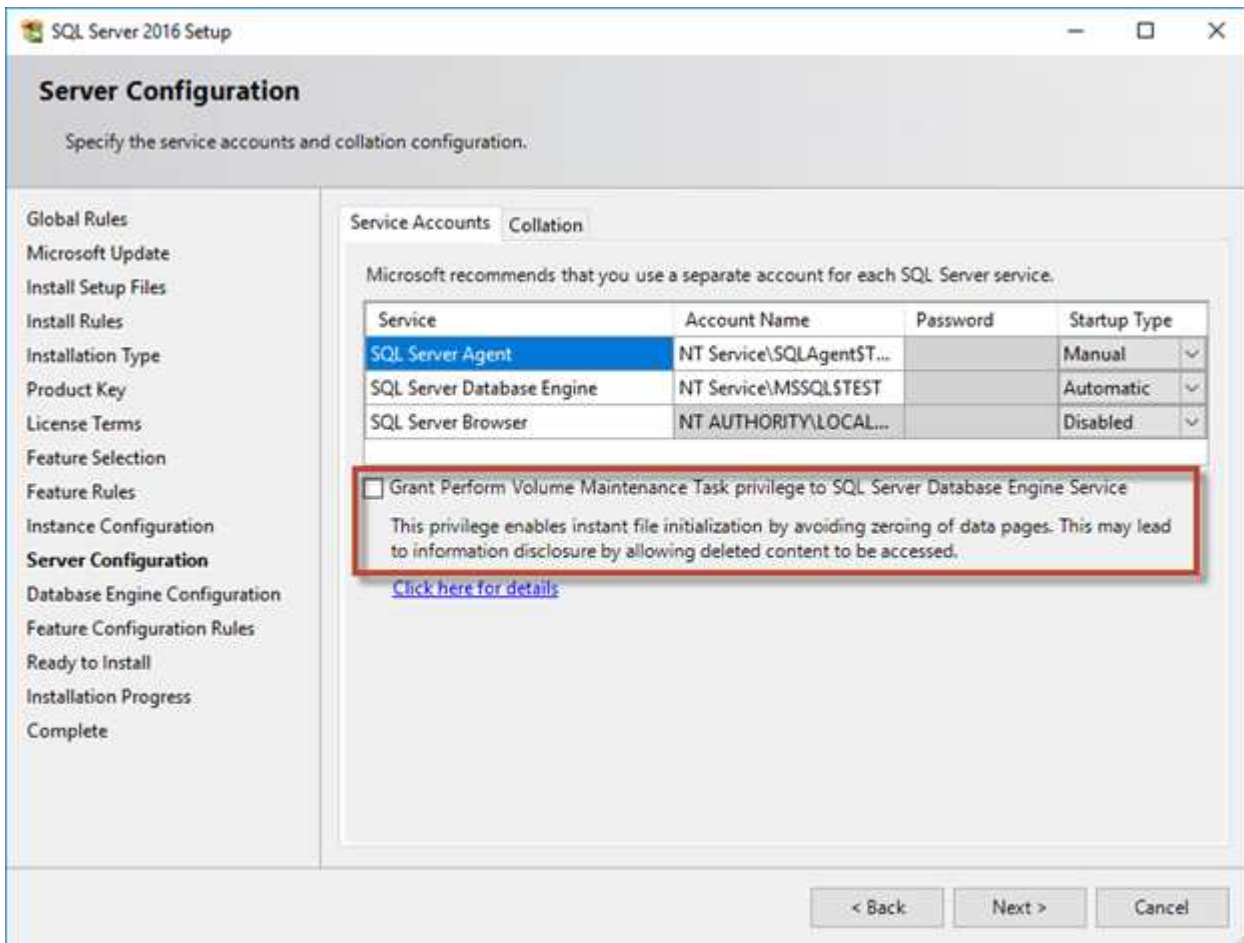
権限が有効になっているかどうかを確認するには、次の例のコードを使用します。このコードは、SQL Serverがエラーログに追加情報を書き込み、小さなデータベースを作成し、ログの内容を読み取るように強制する2つのトレースフラグを設定します。

```
DBCC TRACEON(3004,3605,-1)
GO
CREATE DATABASE DelMe
GO
EXECUTE sp_readerrorlog
GO
DROP DATABASE DelMe
GO
DBCC TRACEOFF(3004,3605,-1)
GO
```

インスタントファイルの初期化が有効になっていない場合、次の例に示すように、SQL Serverのエラーログには、LDFログファイルの初期化に加えてMDFデータファイルが初期化されていることが示されます。インスタントファイルの初期化を有効にすると、ログファイルの初期化のみが表示されます。

| | LogDate | ProcessInfo | Text |
|-----|-------------------------|-------------|--|
| 365 | 2017-02-09 08:10:07.660 | spid53 | Ckpt dbid 3 flush delta counts. |
| 366 | 2017-02-09 08:10:07.660 | spid53 | Ckpt dbid 3 logging active xact info. |
| 367 | 2017-02-09 08:10:07.750 | spid53 | Ckpt dbid 3 phase 1 ended (8) |
| 368 | 2017-02-09 08:10:07.750 | spid53 | About to log Checkpoint end. |
| 369 | 2017-02-09 08:10:07.880 | spid53 | Ckpt dbid 3 complete |
| 370 | 2017-02-09 08:10:08.130 | spid53 | Starting up database 'DelMe'. |
| 371 | 2017-02-09 08:10:08.150 | spid53 | FixupLog Tail(progress) zeroing C:\Program Files\Micros... |
| 372 | 2017-02-09 08:10:08.160 | spid53 | Zeroing C:\Program Files\Microsoft SQL Server\MSSQ... |
| 373 | 2017-02-09 08:10:08.170 | spid53 | Zeroing completed on C:\Program Files\Microsoft SQL... |
| 374 | 2017-02-09 08:10:08.710 | spid53 | Ckpt dbid 6 started |
| 375 | 2017-02-09 08:10:08.710 | spid53 | About to log Checkpoint begin. |

ボリュームメンテナンスタスクはSQL Server 2016では簡単に実行でき、インストールプロセス中にオプションとして提供されます。この図は、SQL Serverデータベースエンジンサービスにボリュームメンテナンスタスクを実行する権限を付与するオプションを示しています。



データベースファイルのサイズを制御するもう1つの重要なデータベースオプションは、自動縮小です。このオプションを有効にすると、SQL Serverはデータベースファイルを定期的に縮小してサイズを縮小し、オペレーティングシステムにスペースを解放します。この処理はリソースを大量に消費するため、新しいデータがシステムに入ってくるとデータベースファイルが再び拡張されるため、あまり有用ではありません。データベースで自動縮小を有効にしないでください。

Microsoft SQL Serverログディレクトリ

ログディレクトリは、トランザクションログバックアップデータをホストレベルで格納するためにSQL Serverで指定します。SnapCenterを使用してログファイルをバックアップする場合は、SnapCenterで使用される各SQL Serverホストに、ログバックアップを実行するようにホストログディレクトリを設定する必要があります。SnapCenterにはデータベースリポジトリがあるため、バックアップ、リストア、クローニングの処理に関連するメタデータは中央のデータベースリポジトリに格納されます。

ホストログディレクトリのサイズは、次のように計算されます。ホストログディレクトリのサイズ= ((最大DB LDFサイズx日次ログ変更率%) x (Snapshot保持率)) ÷ (1 LUNオーバーヘッドスペース%) ホストログディレクトリのサイジング式では、LUNオーバーヘッドスペースが10%であることを前提としています。

ログディレクトリは専用のボリュームまたはLUNに配置します。ホストログディレクトリのデータ量は、バックアップのサイズとバックアップを保持する日数によって異なります。SnapCenterでは、SQL Serverホストごとに1つのホストログディレクトリのみが許可されます。ホストログディレクトリは、SnapCenter → ホスト → プラグインの設定で設定できます。

- NetAppでは、ホストログディレクトリに次のことを推奨しています*。
- ホストログディレクトリが、バックアップSnapshotデータを破損する可能性のある他のタイプのデータと共有されていないことを確認してください。
- マウントポイントをホストするLUNにユーザデータベースまたはシステムデータベースを配置しないでください。
- SnapCenterによるトランザクション・ログのコピー先となる専用のFlexVolボリューム上に、ホスト・ログ・ディレクトリを作成します。
- SnapCenterウィザードを使用してデータベースをNetAppストレージに移行し、データベースを有効な場所に格納できるようにすることで、SnapCenterのバックアップおよびリストア処理を正常に実行できます。移行プロセスはシステムの停止を伴うため、移行の実行中にデータベースを原因でオフラインにする可能性があることに注意してください。
- SQL Serverのフェイルオーバークラスティンスタンス (FCI) では、次の条件が満たされている必要があります。
 - フェイルオーバークラスティンスタンスを使用している場合は、ホストログディレクトリLUNがSnapCenter、バックアップ対象のSQL Serverインスタンスと同じクラスタグループ内のクラスタディスクリソースである必要があります。
 - フェイルオーバークラスティンスタンスを使用している場合は、SQL Serverインスタンスに関連付けられたクラスタグループに割り当てられた物理ディスククラスタリソースである共有LUNにユーザデータベースを配置する必要があります。



Microsoft SQL Server tempdbファイル

tempdbデータベースは大量に利用できます。ONTAPへのユーザデータベースファイルの最適な配置に加えて、tempdbデータファイルを変更して割り当ての競合を軽減

ページ競合は、SQL Serverが新しいオブジェクトを割り当てるために特別なシステムページに書き込む必要がある場合に、グローバル割り当てマップ (GAM)、共有グローバル割り当てマップ (SGAM)、またはページ空きスペース (PFS) ページで発生する可能性があります。ラッチは、メモリ内のこれらのページを保護 (ロック) します。ビジー状態のSQL Serverインスタンスでは、tempdbのシステムページでラッチを取得す

るのに時間がかかることがあります。その結果、クエリの実行時間が長くなり、ラッチ競合と呼ばれます。tempdbデータファイルを作成する場合は、次のベストプラクティスを参照してください。

- 8コア以下の場合：tempdbデータファイル=コア数
- 8コアを超える場合：8個のtempdbデータファイル

次のスクリプト例は、8つのtempdbファイルを作成し、tempdbをマウントポイントに移動することで、tempdbを変更します。C:\MSSQL\tempdb（SQL Server 2012以降）。

```
use master

go

-- Change logical tempdb file name first since SQL Server shipped with
logical file name called tempdev

alter database tempdb modify file (name = 'tempdev', newname =
'tempdev01');

-- Change location of tempdev01 and log file

alter database tempdb modify file (name = 'tempdev01', filename =
'C:\MSSQL\tempdb\tempdev01.mdf');

alter database tempdb modify file (name = 'templog', filename =
'C:\MSSQL\tempdb\templog.ldf');

GO

-- Assign proper size for tempdev01

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'tempdev01', SIZE = 10GB );

ALTER DATABASE [tempdb] MODIFY FILE ( NAME = N'templog', SIZE = 10GB );

GO

-- Add more tempdb files

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev02', FILENAME =
N'C:\MSSQL\tempdb\tempdev02.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev03', FILENAME =
N'C:\MSSQL\tempdb\tempdev03.ndf' , SIZE = 10GB , FILEGROWTH = 10%);
```

```

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev04', FILENAME =
N'C:\MSSQL\tempdb\tempdev04.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev05', FILENAME =
N'C:\MSSQL\tempdb\tempdev05.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev06', FILENAME =
N'C:\MSSQL\tempdb\tempdev06.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev07', FILENAME =
N'C:\MSSQL\tempdb\tempdev07.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

ALTER DATABASE [tempdb] ADD FILE ( NAME = N'tempdev08', FILENAME =
N'C:\MSSQL\tempdb\tempdev08.ndf' , SIZE = 10GB , FILEGROWTH = 10%);

GO

```

SQL Server 2016以降では、インストール時にオペレーティングシステムが認識できるCPUコアの数が自動的に検出され、その数に基づいて、最適なパフォーマンスを実現するために必要なtempdbファイルの数が計算および設定されます。

Microsoft SQL ServerとStorage Efficiency

ONTAPのStorage Efficiency機能は、消費するストレージスペースが最小限に抑えられ、システム全体のパフォーマンスにほとんど影響しないようにSQL Serverデータを格納、管理できるように最適化されています。

Storage Efficiencyは、RAID、プロビジョニング（全体的なレイアウトと利用率）、ミラーリング、その他のデータ保護テクノロジーを組み合わせたものです。Snapshot、シンプロビジョニング、クローニングなどのNetAppテクノロジーは、インフラ内の既存のストレージを最適化し、将来のストレージ支出を先送りまたは回避します。これらのテクノロジーを組み合わせるほど、削減効果が大きくなります。

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

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



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

圧縮

オールフラッシュストレージシステムが登場する以前は、アレイベースの圧縮の価値は限られていました。I/O負荷の高いワークロードのほとんどでは、許容可能なパフォーマンスを提供するために非常に多数のスピンダルが必要だったためです。ストレージシステムには、ドライブ数が多いことの副作用として、必要以上の容量が常に搭載されていました。この状況は、ソリッドステートストレージの登場によって変化しました。

た。優れたパフォーマンスを得るためだけにドライブを過剰にオーバプロビジョニングする必要はもうありません。ストレージシステムのドライブスペースは、実際の容量ニーズに合わせて調整できます。

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

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

適応圧縮

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

ONTAPは物理ブロックを4KB単位で管理します。アダプティブ圧縮では、デフォルトの圧縮ブロックサイズである8KBが使用されます。つまり、データは8KB単位で圧縮されます。これは、リレーショナルデータベースで最もよく使用される8KBのブロックサイズに一致します。圧縮アルゴリズムは、より多くのデータが1つの単位として圧縮されるので、より効率的になります。圧縮ブロックサイズが32KBの場合、8KBの圧縮ブロックユニットよりもスペース効率に優れています。つまり、デフォルトの8KBのブロックサイズを使用するアダプティブ圧縮の場合、削減率はわずかに低くなりますが、圧縮ブロックサイズを小さくすることには大きなメリットがあります。データベースワークロードには、大量の上書きアクティビティが含まれています。32KBの圧縮されたデータブロックの8KBを上書きするには、32KBの論理データ全体を読み取って解凍し、必要な8KB領域を更新してから再度圧縮し、32KB全体をドライブに書き込む必要があります。この処理はストレージシステムでは非常にコストがかかります。このため、圧縮ブロックサイズの大きい競合ストレージアレイでも、データベースワークロードのパフォーマンスが大幅に低下します。



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



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

圧縮のアライメント

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

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

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

データコンパクション

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

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

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

温度に基づくストレージ効率

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

重複排除

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

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

一部のデータに重複データが含まれていません。たとえば、Oracleブロックには、データベースに対してグローバルに一意的なヘッダーと、ほぼ一意的なトレーラが含まれています。そのため、Oracleデータベースの重複排除によって1%以上の削減効果が得られることはほとんどありません。MS SQLデータベースでの重複排除は

やや優れていますが、ブロックレベルでの固有のメタデータは依然として制限されています。

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

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

効率性とシンプロビジョニング

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

シンプロビジョニングは、管理を簡易化しながら、使用可能な容量を大幅に改善し、コストを削減できるため、強く推奨されます。これは、単純なデータベース環境では、多くの場合、空のスペース、多数のボリュームやLUN、圧縮可能なデータが含まれているためです。シックプロビジョニングでは、ボリュームとLUNのストレージにスペースがリザーブされます。これは、100%フルになり、100%圧縮不可能なデータが含まれる場合に限られます。これは起こりそうもないことです。シンプロビジョニングを使用すると、スペースを他の場所で再生して使用できます。また、容量の管理は、多数の小さいボリュームやLUNではなく、ストレージシステム自体に基づいて行うことができます。

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

*注意：*ボリュームがシックプロビジョニングされている場合は、ボリュームの圧縮解除や重複排除の削除など、そのボリュームのすべての効率化機能を完全に無効にするように注意する必要があります。sis undo コマンドを実行します。ボリュームが volume efficiency show 出力。有効になっている場合、ボリュームはまだ部分的に効率化機能用に設定されています。その結果、オーバーライトギャランティの動作が異なります。そのため、設定で原因が見落とされてボリュームのスペースが予期せず不足し、データベースI/Oエラーが発生する可能性が高くなります。

効率化のベストプラクティス

NetAppの推奨事項は次のとおりです。

AFFのデフォルト

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

一般的な推奨事項

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

データベース圧縮

SQL Server自体には、データを圧縮して効率的に管理する機能もあります。SQL Serverでは現在、行圧縮とページ圧縮の2種類のデータ圧縮がサポートされています。

行圧縮を使用すると、データストレージ形式が変更されます。たとえば、整数と小数を、ネイティブの固定長形式ではなく可変長形式に変更します。また、空白スペースを排除することで、固定長の文字列を可変長形式に変更します。ページ圧縮では、行圧縮と他の2つの圧縮方式（プレフィックス圧縮とディクショナリ圧縮）が実装されます。ページ圧縮の詳細については、"[ページ圧縮の実装](#)"。

データ圧縮は現在、SQL Server 2008以降のEnterprise、Developer、およびEvaluationエディションでサポートされています。圧縮はデータベース自体で実行できますが、SQL Server環境ではほとんど実行されません。

SQL Serverデータファイルのスペース管理の推奨事項は次のとおりです。

- SQL Server環境でシンプロビジョニングを使用すると、スペース利用率を向上し、スペースギャランティ機能を使用する場合に必要なストレージ全体を削減できます。
- ストレージ管理者が監視する必要があるのはアグリゲート内のスペース使用量だけであるため、一般的な構成では自動拡張を使用します。
- バックアップから単一ボリュームへのデータベースのリストアなど、同じデータのコピーがボリュームに複数含まれていることがわかっている場合を除き、SQL Serverデータファイルを含むボリュームで重複排除を有効にしないことを推奨します。

スペース再生

スペース再生は、LUN内の未使用スペースをリカバリするために定期的に開始できます。SnapCenterでは、次のPowerShellコマンドを使用してスペース再生を開始できます。

```
Invoke-SdHostVolumeSpaceReclaim -Path drive_path
```

スペース再生を実行する必要がある場合は、最初にホストのサイクルを消費するため、アクティビティが少ない時間帯にこのプロセスを実行する必要があります。

NetApp管理ソフトウェアによるMicrosoft SQL Serverデータ保護

データベースのバックアップは、ビジネス要件に基づいて計画します。ONTAPのNetApp Snapshotテクノロジーを組み合わせ、Microsoft SQL Server APIを活用することで、ユーザデータベースのサイズに関係なく、アプリケーションと整合性のあるバックアップを迅速に作成できます。より高度なデータ管理やスケールアウトデータ管理の要件に対応するために、NetAppはSnapCenterを提供しています。

SnapCenter

SnapCenterは、エンタープライズアプリケーション向けのNetAppデータ保護ソフトウェアです。SnapCenter Plug-in for SQL Serverや、SnapCenter Plug-in for Microsoft Windowsで管理されるOS処理を使用して、SQL Serverデータベースを迅速かつ簡単に保護できます。

SQL Serverインスタンスは、スタンドアロンセットアップ、フェイルオーバークラスティンスタンス、または常時稼働の可用性グループにすることができます。その結果、データベースの保護、クローニング、リストアをプライマリコピーまたはセカンダリコピーから単一コンソールで実行できます。SnapCenterでは、SQL Serverデータベースをオンプレミス、クラウド、ハイブリッド構成の両方で管理できます。また、開発やレポート作成のために、元のホストまたは代替ホストに数分でデータベースコピーを作成することもできます。



* NetAppでは* SnapCenterを使用してSnapshotコピーを作成することを推奨しています。以下に示すT-SQL方式も機能しますが、SnapCenterでは、バックアップ、リストア、クローニングのプロセスを完全に自動化できます。また、検出を実行して、正しいSnapshotが作成されていることを確認します。事前設定は必要ありません。な...何だ? また、SQL Serverでは、作成時にSnapshotに正しいデータが存在するように、OSとストレージの間で調整を行う必要があります。ほとんどの場合、これを行う唯一の安全な方法は、SnapCenterまたはT-SQLを使用することです。この追加の調整なしで作成されたSnapshotは、確実にリカバリできない可能性があります。

SQL Server Plug-in for SnapCenterの詳細については、を参照してください。"[TR-4714 : 『Best Practice Guide for SQL Server using NetApp SnapCenter』](#)".

T-SQLスナップショットを使用したデータベースの保護

SQL Server 2022では、MicrosoftがT-SQLスナップショットを導入しました。これにより、バックアップ処理のスクリプト作成と自動化を行うことができます。フルサイズのコピーを実行する代わりに、Snapshot用にデータベースを準備できます。データベースのバックアップ準備が完了したら、ONTAP REST APIを使用してSnapshotを作成できます。

次に、バックアップワークフローの例を示します。

1. ALTERコマンドを使用してデータベースをフリーズします。これにより、基盤となるストレージ上で整合性のあるSnapshotを作成するためのデータベースが準備されます。フリーズ後、backupコマンドを使用してデータベースをフリーズ解除し、スナップショットを記録できます。
2. 新しいbackup groupコマンドとbackup serverコマンドを使用して、ストレージボリューム上の複数のデータベースのスナップショットを同時に実行します。
3. フルバックアップまたはCOPY_ONLYフルバックアップを実行します。これらのバックアップもmsdbに記録されます。

4. スナップショットフルバックアップ後に通常のストリーミング方式で作成されたログバックアップを使用して、ポイントインタイムリカバリを実行します。必要に応じて、ストリーミング差分バックアップもサポートされます。

詳細については、を参照してください "[T-SQLスナップショットについて知るためのMicrosoftのドキュメント](#)"。

ONTAPを使用したMicrosoft SQL Serverディザスタリカバリ

エンタープライズデータベースやアプリケーションインフラでは、自然災害や予期しないビジネスの中断からダウンタイムを最小限に抑えて保護するために、レプリケーションが必要になることがよくあります。

SQL Server Always-On可用性グループレプリケーション機能は優れたオプションであり、NetAppはデータ保護とAlways-Onを統合するオプションを提供します。ただし、ONTAPレプリケーションテクノロジーを検討する必要がある場合もあります。MetroClusterやSnapMirrorなどのONTAPレプリケーションオプションを使用すると、パフォーマンスへの影響を最小限に抑えながら拡張性が向上し、SQL以外のデータを保護できます。また、一般にインフラ全体のレプリケーションとDR解決策が提供されます。

SnapMirror非同期

SnapMirrorテクノロジーは、LANおよびWAN経由でデータをレプリケートするための、高速で柔軟な非同期エンタープライズ解決策を提供します。最初のミラーリングの作成後は、変更されたデータブロックのみがデスティネーションに転送されるため、必要なネットワーク帯域幅が大幅に削減されます。

SnapMirror for SQL Serverの推奨事項は次のとおりです。

- CIFSを使用する場合は、デスティネーションSVMがソースSVMと同じActive Directoryドメインのメンバーである必要があります。これにより、NASファイルに格納されているアクセス制御リスト (ACL) が災害からのリカバリ時に破損しないようになります。
- ソースボリューム名と同じデスティネーションボリューム名を使用する必要はありませんが、デスティネーションボリュームをデスティネーションにマウントするプロセスを管理しやすくすることができます。CIFSを使用する場合は、デスティネーションNASネームスペースをソースネームスペースとパスおよびディレクトリ構造で同一にする必要があります。
- 整合性を確保するために、コントローラからのSnapMirror更新のスケジュールを設定しないでください。代わりに、フルバックアップまたはログバックアップの完了後にSnapCenterからのSnapMirror更新を有効にしてSnapMirrorを更新します。
- SQL Serverデータを含むボリュームをクラスタ内の複数のノードに分散して、すべてのクラスタノードでSnapMirrorレプリケーションアクティビティを共有できるようにします。この分散により、ノードリソースの使用が最適化されます。

SnapMirrorの詳細については、を参照してください。 "[TR-4015：『SnapMirrorの設定およびベストプラクティスガイド- ONTAP 9』](#)"。

ONTAP上のMicrosoft SQL Serverのセキュリティ保護

SQL Serverデータベース環境のセキュリティ保護は、データベース自体の管理にとどまらない多角的な取り組みです。ONTAPには、データベースインフラのストレージを保

護するために設計された独自の機能がいくつか用意されています。

Snapshot コピー

ストレージスナップショットは、ターゲットデータのポイントインタイムレプリカです。ONTAPには、さまざまなポリシーを設定し、ボリュームあたり最大1024個のSnapshotを格納する機能が実装されています。ONTAPのSnapshotはスペース効率に優れています。スペースが消費されるのは、元のデータセットが変更されたときだけです。また、読み取り専用です。Snapshotは削除できますが、変更することはできません。

場合によっては、ONTAPで直接Snapshotのスケジュールを設定できます。また、スナップショットを作成する前に、SnapCenterなどのソフトウェアでアプリケーションやOSの処理をオーケストレーションしなければならない場合もあります。ワークロードに最適なアプローチにかかわらず、アグレッシブなスナップショット戦略を使用すると、ブートLUNからミッションクリティカルなデータベースまで、すべてのバックアップに頻繁かつ簡単にアクセスできるため、データのセキュリティを確保できます。

注：ONTAPフレキシブルボリューム、つまり、ボリュームはLUNと同義ではありません。ボリュームは、ファイルやLUNなどのデータ用の管理コンテナです。たとえば、データベースを8 LUNのストライプセットに配置し、すべてのLUNを1つのボリュームに格納するとします。

スナップショットの詳細については、["こちらをご覧ください。"](#)

スナップショットの改ざん防止

ONTAP 9.12.1以降では、Snapshotは読み取り専用であるだけでなく、偶発的または意図的な削除からも保護できます。この機能は改ざん防止スナップショットと呼ばれます。保持期間は、Snapshotポリシーを使用して設定および適用できます。作成されたスナップショットは、有効期限に達するまで削除できません。管理またはサポートセンターのオーバーライドはありません。

これにより、侵入者、悪意のある内部者、さらにはランサムウェア攻撃さえも、バックアップが原因でONTAPシステム自体にアクセスできたとしても、バックアップを侵害することはできません。頻繁なSnapshotスケジュールと組み合わせることで、非常に強力なデータ保護と非常に低いRPOを実現できます。

改ざん防止スナップショットの詳細については、["こちらをご覧ください。"](#)

SnapMirror レプリケーション

Snapshotはリモートシステムにレプリケートすることもできます。これには改ざん防止Snapshotも含まれます。このSnapshotでは、リモートシステムに保持期間が適用され、適用されます。その結果、ローカルSnapshotと同じデータ保護のメリットが得られますが、データは2つ目のストレージレイに配置されます。これにより、元のレイを破棄してもバックアップが損なわれることはありません。

2つ目のシステムでは、管理セキュリティのための新しいオプションも開きます。たとえば、NetAppのお客様によっては、プライマリストレージシステムとセカンダリストレージシステムの認証クレデンシャルを分離している場合があります。1人の管理ユーザが両方のシステムにアクセスできることはないため、悪意のある管理者がデータのすべてのコピーを削除することはできません。

SnapMirrorの詳細については、["こちらをご覧ください。"](#)

Storage Virtual Machine

新しく構成したONTAPストレージシステムは、新しくプロビジョニングしたVMware ESXサーバと似ています。これは、仮想マシンが作成されるまで、どちらもユーザをサポートできないためです。ONTAPでは、ストレージ管理の最も基本的な単位となるStorage Virtual Machine (SVM) を作成します。各SVMには、独自のストレージリソース、プロトコル構成、IPアドレス、FCP WWNがあります。これがONTAPマルチテナンシーの基盤です。

たとえば、重要な本番環境のワークロード用に1つのSVMを設定し、開発アクティビティ用にもう1つのSVMを別のネットワークセグメントに設定できます。これにより、本番用SVMへのアクセスを特定の管理者に制限し、開発者は開発用SVM内のストレージリソースをより広範に制御できるようになります。また、特に重要な目に見えるデータを格納するために、財務チームや人事チームに3つ目のSVMを用意しなければならない場合もあります。

SVMの詳細については、["こちらをご覧ください。"](#)

カンリRBAC

ONTAPには、管理者ログイン用の強力なロールベースアクセス制御 (RBAC) が用意されています。クラスターへのフルアクセスが必要な管理者もいれば、特定のSVMへのアクセスのみが必要な管理者もいます。高度なヘルプデスク担当者は、ボリュームのサイズを増やす機能を必要とする場合があります。その結果、管理者ユーザに、職務を遂行するために必要なアクセス権を付与するだけでなく、それ以上のアクセス権を付与することができます。さらに、さまざまなベンダーのPKIを使用してこれらのログインを保護し、sshキーのみへのアクセスを制限し、失敗したログイン試行のロックアウトを強制できます。

管理アクセス制御の詳細については、["こちらをご覧ください。"](#)

マルチファクタ認証

ONTAPおよびその他の一部のNetApp製品では、さまざまな方法を使用した多要素認証 (MFA) がサポートされるようになりました。その結果、ユーザー名/パスワードだけでは、FOBやスマートフォンアプリなどの2番目の要因からのデータがないセキュリティスレッドではありません。

詳細については、をクリックしてください ["こちらをご覧ください。"](#)

API RBAC

自動化にはAPI呼び出しが必要ですが、すべてのツールにフル管理アクセスが必要となるわけではありません。自動化システムを保護するために、APIレベルでRBACを使用することもできます。自動化ユーザアカウントは、必要なAPI呼び出しだけに制限できます。たとえば、監視ソフトウェアには変更アクセスは必要なく、読み取りアクセスのみが必要です。ストレージをプロビジョニングするワークフローでは、ストレージを削除する機能は必要ありません。

詳細については、[c here](#).

マルチ管理者認証 (MAV)

特定のアクティビティを承認するために、それぞれ独自のクレデンシャルを持つ2人の異なる管理者を要求することで、多要素認証をさらに進めることができます。これには、ログイン権限の変更、診断コマンドの実行、データの削除が含まれます。

マルチ管理者認証 (MAV) の詳細については、["こちらをご覧ください"](#)

著作権に関する情報

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5252.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および/または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用权を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用权については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。