



テクニカル レポート

ONTAPを使用したMicrosoft SQL Serverの ベストプラクティスガイド

NetApp
Manohar Kulkarni / Pat Sinthusan
2022年5月 | TR-4590

概要

このベストプラクティスガイドでは、ストレージ管理者およびデータベース管理者がMicrosoft SQL ServerをNetApp® ストレージに導入する方法について説明します。

<<本レポートは機械翻訳による参考訳です。公式な内容はオリジナルである英語版をご確認ください。>>

目次

はじめに.....	4
目的と範囲	4
対象読者.....	4
SQL Serverのワークロードタイプ.....	4
ベンチマークのテスト	5
SQL Server構成	5
共有インスタンスと専用インスタンス	5
CPU構成	5
メモリ構成	7
データベースファイルとファイルグループ.....	12
データベースとストレージに関する考慮事項.....	16
アグリゲート.....	16
ボリューム	16
LUN	17
データストレージの設計.....	19
ストレージ効率と管理性.....	19
Snapshotコピー	19
シンプロビジョニング	20
スペースギャランティ	20
スペース再生.....	20
フラクショナルリザーブ.....	21
NetApp FlexClone	22
圧縮、コンパクション、重複排除.....	22
NetApp SnapMirror	24
NetAppクラウドデータサービスとSQL Server	25
まとめ	25
追加情報の入手方法.....	25

表一覧

表1) 一般的なSQL Serverワークロード	5
表2) ボリュームギャランティをnoneに設定	21
表3) 自動削除と自動拡張を使用したボリュームのセットアップ	21

図一覧

図1) 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す	6
図2) SQL Server Management Studio を使用したサーバメモリの最小値と最大値の調整	8
図3) SQL Server Management Studio を使用した最大ワーカースレッド数の設定.....	9
図4) SQL Server Management Studioを使用したインデックス作成メモリとクエリごとの最小メモリの設定...	11
図5) ファイルグループ	13
図6) SQL Serverのインストール時にボリュームメンテナンスタスクの実行権限を付与するオプション.....	15

はじめに

SQL ServerはMicrosoftのデータプラットフォームの基盤であり、オンプレミスでもクラウドでも、インメモリテクノロジーを使用してミッションクリティカルなパフォーマンスを実現し、あらゆるデータの分析を高速化します。Microsoft SQL Serverは、ミッションクリティカルなアプリケーションに画期的なパフォーマンス、可用性、管理性を提供することで、以前のリリースで提供されていたミッションクリティカルな機能を基盤としています。ストレージシステムは、SQL Serverデータベースの全体的なパフォーマンスの重要な要素です。NetAppは、SQL Serverデータベースでエンタープライズクラスのパフォーマンスを実現しながら、環境を管理するためのワールドクラスのツールを提供するための製品を複数提供しています。

モクテキトモクテキ

このドキュメントでは、効果的かつ効率的なストレージ導入とエンドツーエンドのデータ保護および保持計画を実現するために、NetApp ONTAP®ソフトウェアを実行するNetAppストレージシステムにSQL Serverを導入する際のベストプラクティスと設計上の考慮事項について説明します。このガイドの範囲は、SQL Serverの導入時にNetAppが推奨するストレージインフラの設計原則と推奨基準に基づいた技術設計ガイドラインに限定されます。エンドツーエンドの実装は、このレポートの範囲外です。

このガイドで説明するベストプラクティスと推奨事項により、SQL ServerアーキテクトとNetAppストレージ管理者は、可用性が高く管理が容易なSQL Server環境を計画し、厳しいSLAを満たすことができます。NetAppは、読者が次のことを理解していることを前提としています。

- NetApp ONTAPソフトウェア
 - NetApp SnapCenter® ASバックアップソフトウェア：
 - SnapCenter Plug-in for Microsoft Windows
 - SnapCenter Plug-In for SQL Server
- Microsoft SQL Serverのアーキテクチャと管理

NetAppスタック全体での構成の互換性については、[NetApp Interoperability Matrix Tool](#) (IMT) を参照してください。

対象読者

このテクニカルレポートは、お客様の環境にSQL Serverデータベースソリューションを導入する責任を負うNetAppのお客様、パートナー様、従業員、フィールド担当者を対象としています。NetAppは、読者が前述のソリューションのさまざまなコンポーネントに精通していることを前提としています。

SQL Serverワークロードの種類

SQL Serverデータベースプラットフォームでは、複数のアプリケーションをサポートできます。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す。これらのデータベースは通常、お客様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を必要とするSLAがあり、パフォーマンスの低下や可用性に影響を受けやすいものが多くあります。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、ランダム読み取りが75%~90%、書き込みが25%~10%という特徴を持っています。

- 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す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使用率とディスクからの読み取り処理に影響されます。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通常、100%の読み取りI/O構成です。

ベンチマークのテスト

トランザクション処理性能評議会 (TPC) は、トランザクション処理とデータベースのベンチマークを定義し、検証可能で客観的なTPCパフォーマンス データを業界に普及させるために設立された非営利団体です。TPCテストは、ユーザがデータベースに対してトランザクションを実行する完全なコンピューティング環境をシミュレートします。表1に、一般的なSQL Serverのテスト ワークロードを示します。

表1) 一般的なSQL Serverワークロード

ワークロードタイプ	TPCテスト	読み取り/書き込み比率 (割合)
OLTP	TPC-C	~75/25
	TPC-E	~90/10
DSS	TPC-H	~100/0

さまざまなワークロード生成オプションを利用できますが、一般的には、トランザクションワークロードを処理する際のSQL Serverデータベースのパフォーマンス測定に重点を置いており、MicrosoftのTPC-EツールまたはHammerDB ([HammerDB.com](https://hammerdb.com)) を使用したTPC-Hを使用しています。これらの特定のベンチマークの使用方法の詳細については、このドキュメントでは説明しません。

SQL Serverの設定

このセクションでは、SQL Serverのインストール前とインストール後に考慮する必要があるSQL Server設定の構成方法に関する一般的なガイダンスを提供します。

共有インスタンスと専用インスタンス

アプリケーションに多数のスキーマとストアドプロシージャがある場合、SQL Serverインスタンスを共有する他のアプリケーションに影響を与える可能性があります。インスタンスリソースが分割またはロックされる可能性があり、その結果、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。

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る可能性があります。どのインスタンスが根本的な原因かを特定する必要があるためです。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、専用のインスタンスを使用することを強く推奨します。

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。

最適なパフォーマンスを得るために、NetAppではできるだけ専用のSQL Serverインスタンスを選択することを推奨しています。

CPU構成

ハイパースレッディング

ハイパースレッディングは、Intel独自の同時マルチスレッド (SMT) 実装で、x86マイクロプロセッサで実行される計算 (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ドCPUを物理CPUとしてオペレーティング システムに認識できます。

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るようにします。

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のライセンスを取得する際には、主に2つの考え方があります。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インスタンスでコアをすべて同時に使用することはできません。図1に、起動後のSQL Serverログメッセージを示します。このメッセージは、コア制限の適用を示しています。

図1) 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:
-s "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、「[SQL Server 2019 : 最新のデータプラットフォーム](#)」を参照してください。

不均一なメモリアクセス

NUMA (Non-Uniform 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式です。SQL ServerがインストールされているサーバでNUMAを構成する場合は、SQL ServerがNUMAを認識し、NUMAハードウェアで十分に機能するため、追加の構成は必要ありません。

プロセッサアフィニティ

パフォーマンス上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があります。

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に対して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、このような状況が発生するとパフォーマンスに影響を与える可能性があります。この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設定します。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のディスク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を向上させることができます。

並列処理の最大レベル (MAXDOP)

デフォルトでは、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 (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)。これは大規模なクエリには適していますが、パフォーマンスの問題が発生し、同時実行性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数に並列処理を制限する方法が優れています。たとえば、1ソケットあたり12コアの物理CPUソケットが2つあるサーバでは、ハイパースレッ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、1つのバッチクエリで使用できるCPUの数を制限します。

データウェアハウスなどのDSSの場合は、この設定を50ぐらいにしてから、必要に応じて調整することを推奨します。アプリケーション内の重要なクエリを測定し、必要に応じて調整します。

メモリ構成

最大サーバメモリ

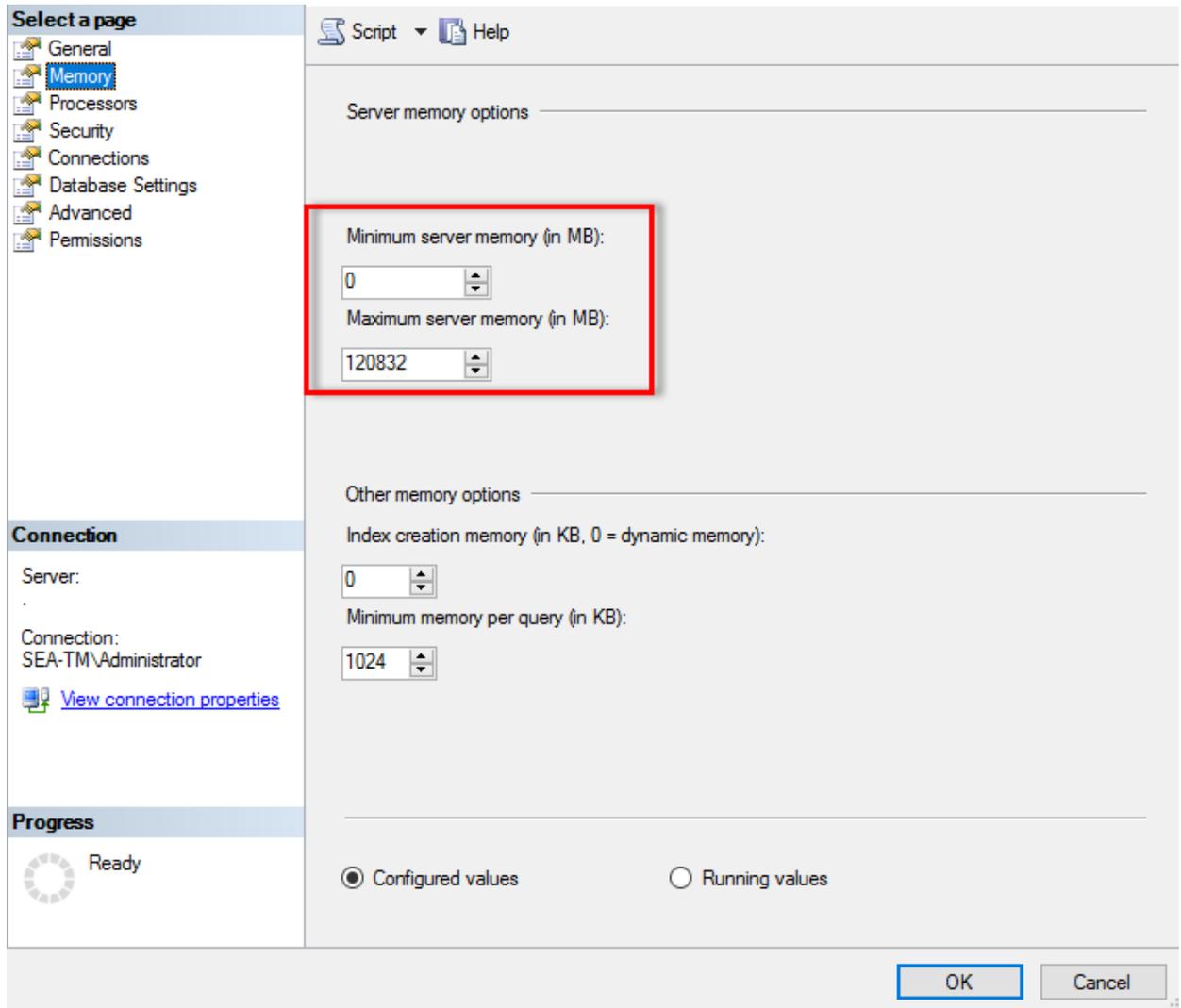
最大サーバメモリ オプションは、SQL Serverインスタンスが使用できるメモリの最大量を設定します。通常、SQL Serverが実行されている同じサーバ上で複数のアプリケーションが実行されている場合に、これらのアプリケーションが正常に機能するために十分なメモリを確保する必要があるときに使用されます。

一部のアプリケーションでは、起動時に使用可能なメモリのみが使用され、必要に応じて要求されない場合があります。ここでは、サーバの最大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に対して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。

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には最低4GBから6GBのRAMを残すことを推奨します。図2は、サーバの最小メモリと最大メモリの設定方法を示しています。

図2) SQL Server Management Studioを使用したサーバメモリの最小値と最大値の調整



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
```

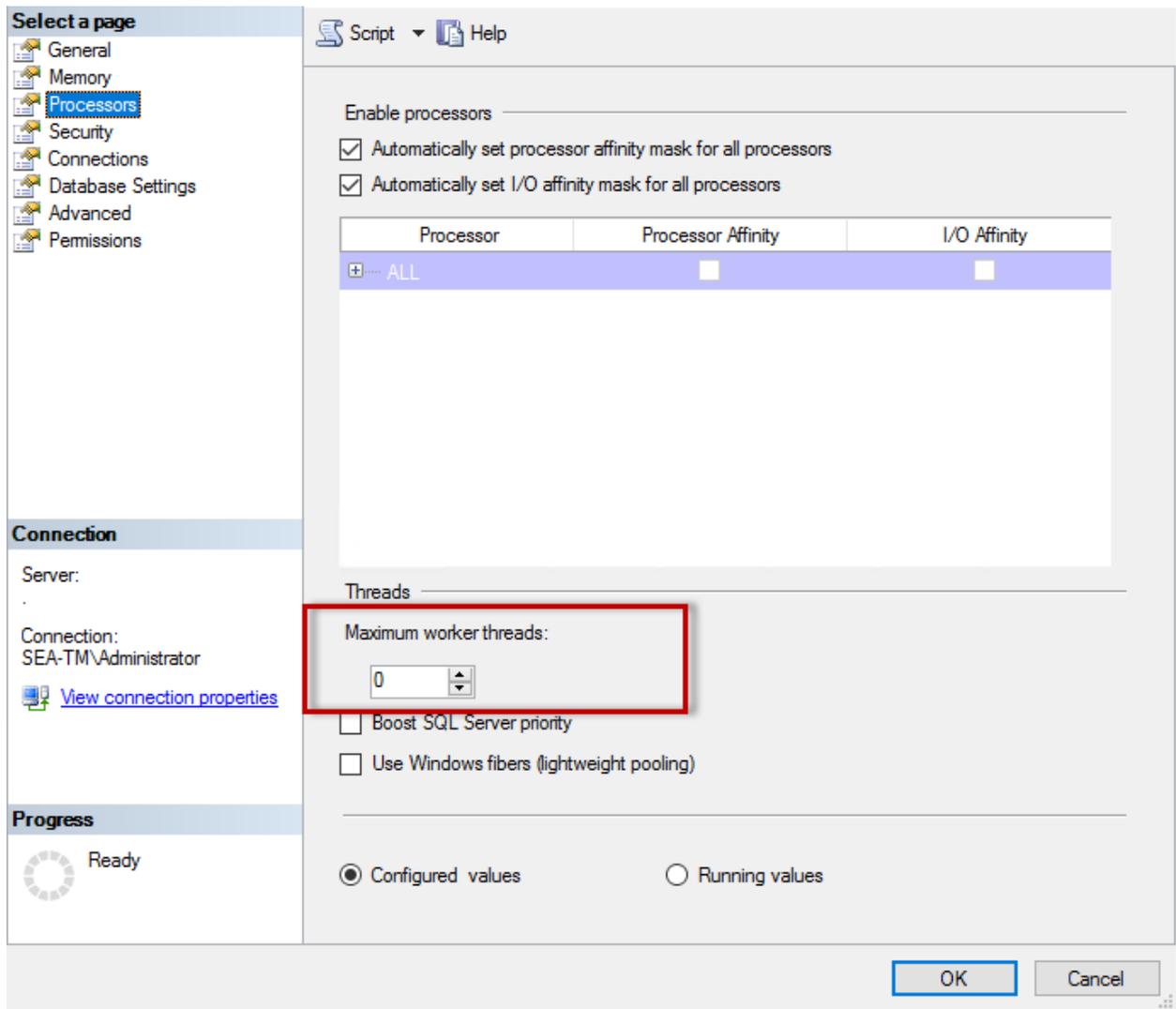
ワーカースレッドの最大数

ワーカー スレッドの最大数オプションを使用すると、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できます。通常、クエリ要求ごとに別々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への同時接続数が数百の場合、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 スレッド数の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、より多く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なので、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を構成する必要があるのは、どのような場合でしょうか？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、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ることがあります。ただし、負荷がCPUに制限されていない場合や、ほかの長い待機時間が発生している場合に限りです。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、[Configure the max worker threads Server Configuration Option](#)を参照してください。図3 は、ワーカー スレッドの最大数を調整する方法を示しています。

図3) SQL Server Management Studioを使用した最大ワーカー スレッド数の設定



次に、T-SQLを使用してmax work threadsオプションを設定する例を示します。

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

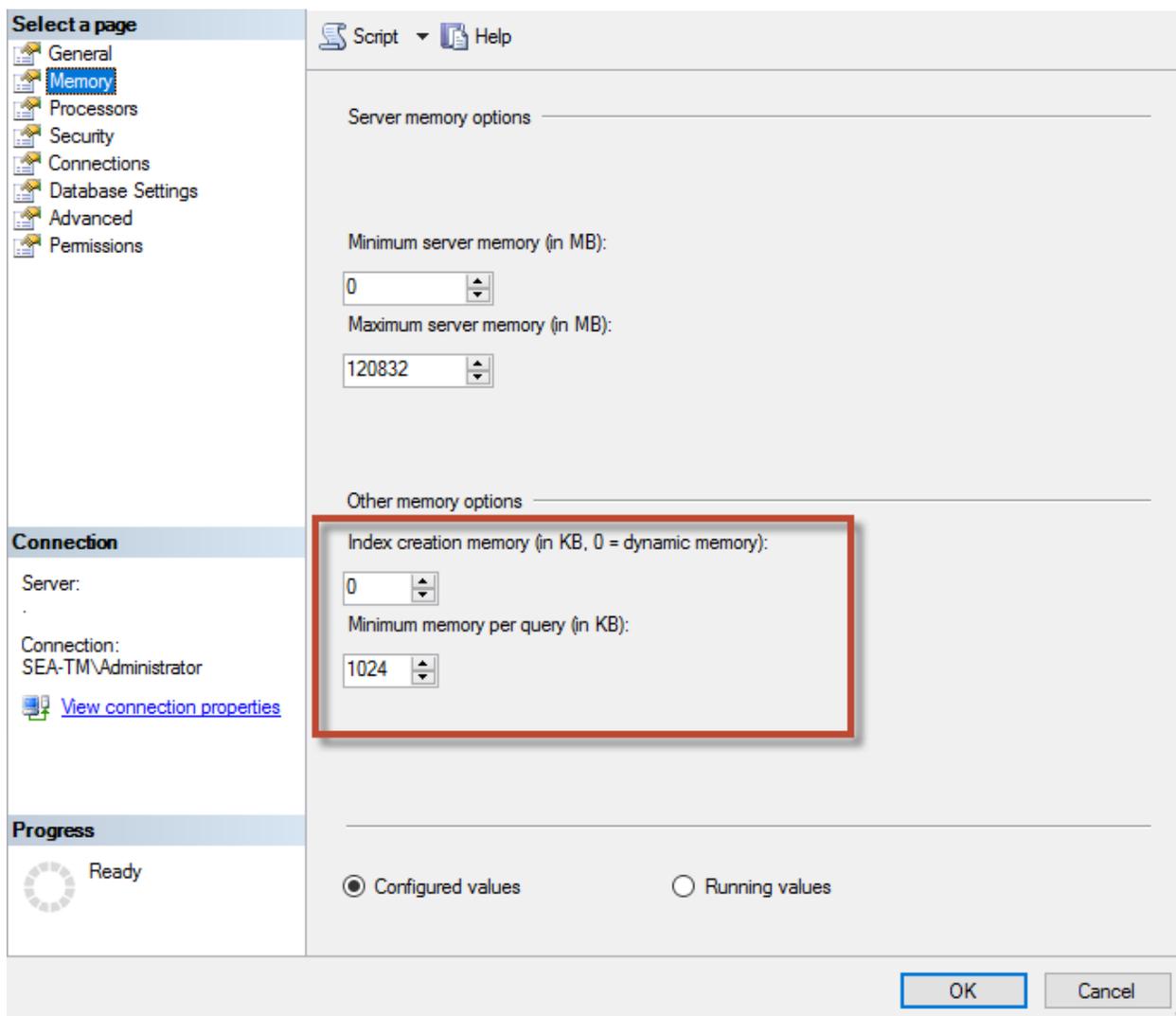
インデックス作成メモリ

`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。インデックスの作成時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で問題が発生した場合は、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、この `min memory per query` 設定では、実行するクエリごとに>または=が1024KBに割り当てられます。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、この設定を強化することで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。

図4) SQL Server Management Studioを使用したインデックス作成メモリとクエリごとの最小メモリの設定



バッファプール拡張

バッファプール拡張機能は、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ます。バッファプール拡張機能は、SQL Serverのエディションによっては使用できない場合があります。64ビットのSQL Server Standard、Business Intelligence、Enterprise Edition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パフォーマンスが高いため、バッファプールの拡張によって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レイテンシの削減
- トランザクションスループットの向上

- より大規模なハイブリッド バッファ プールによる、読み取りパフォーマンスの向上
- バッファプール拡張のために既存および将来の低コストメモリを利用できるキャッシュアーキテクチャである

NetAppでは、次の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、それより大きい必要があります。

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
```

データベースファイルとファイルグループ

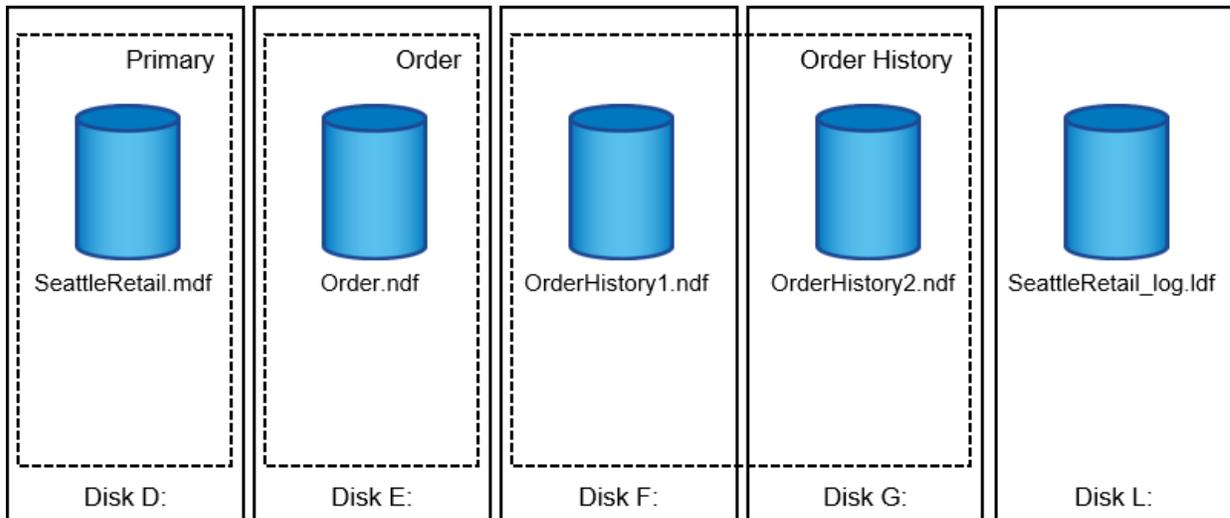
SQL Serverデータベースは、データの格納と操作を可能にするオブジェクトの集まりです。理論的には、SQL Server（64ビット）ではインスタンスあたり32,767個のデータベースと524,272TBのデータベース サイズがサポートされますが、一般的なインストールでは複数のデータベースが使用されています。ただし、SQL Serverで処理できるデータベースの数は、負荷とハードウェアによって異なります。SQL Serverインスタンスでは、数十、数百、場合によっては数千の小規模データベースをホストしていることも珍しくありません。

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、アクションを元に戻す（ロールバックする）か、やり直す（リプレイする）かを指定します。SQL Serverのトランザクションログは、データの整合性と堅牢性に対する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。すべてのデータ操作言語（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、プライマリ データ ファイルが1つあります。デフォルトでは、.mdf拡張子が付いています。また、各データベースにはセカンダリ データベース ファイルを含めることができます。デフォルトでは、これらのファイルには.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、データベースの管理を簡易化します。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 ファイルの構成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。

図5) ファイルグループ



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に行われます。

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たデータ量を書き込みます。ファイル内の空きスペースが多いほど、処理される書き込み量も多くなります。

NetAppでは、単一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。

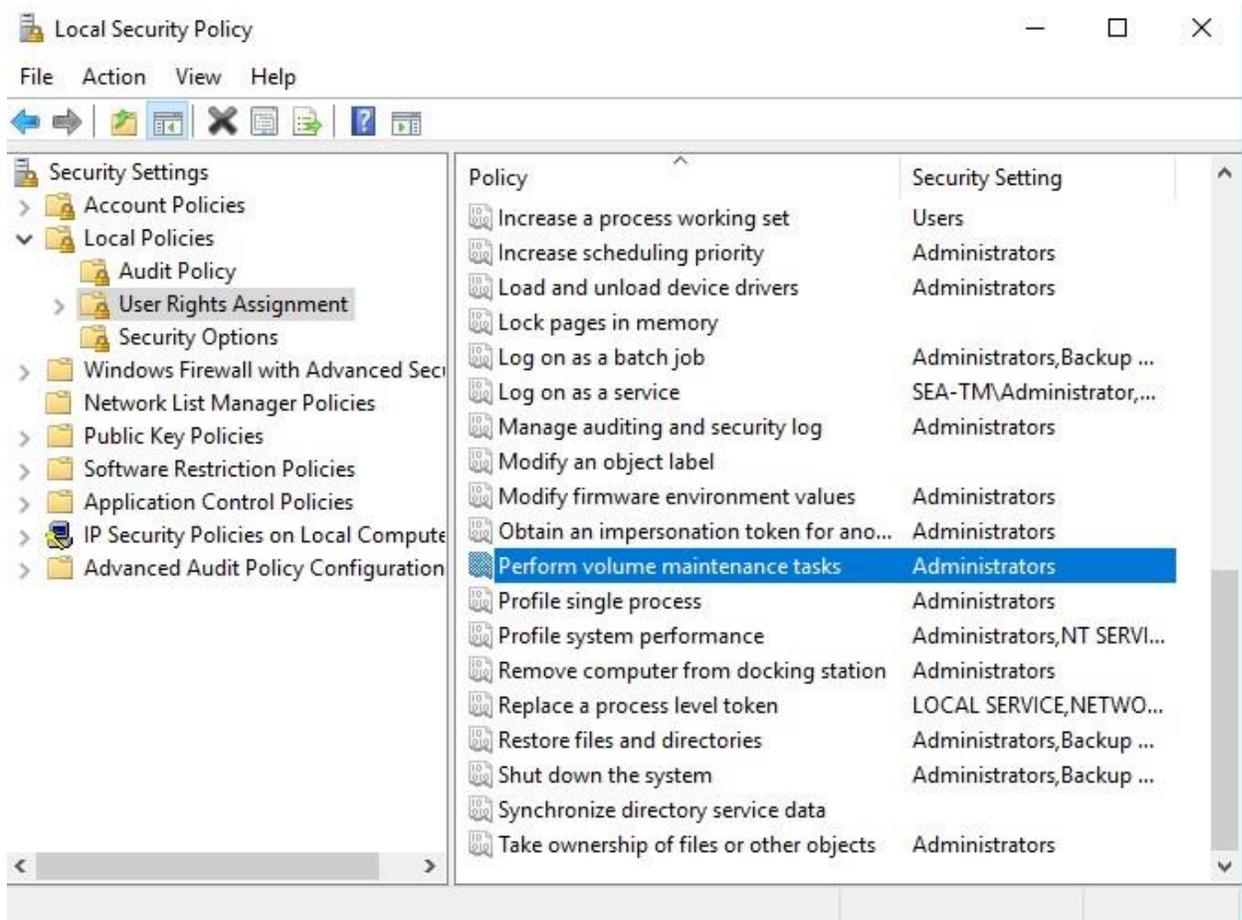
SQL Serverは、ファイルを拡張するたびに、新しく割り当てられたスペースをゼロで埋めます。このプロセスは、対応するファイルに書き込む必要があるすべてのセッションをブロックします。トランザクションログが増加した場合は、トランザクションログレコードを生成します。

SQL Serverは常にトランザクションログをゼロにします。この動作は変更できません。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、セキュリティ上の小さなリスクが伴います。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。

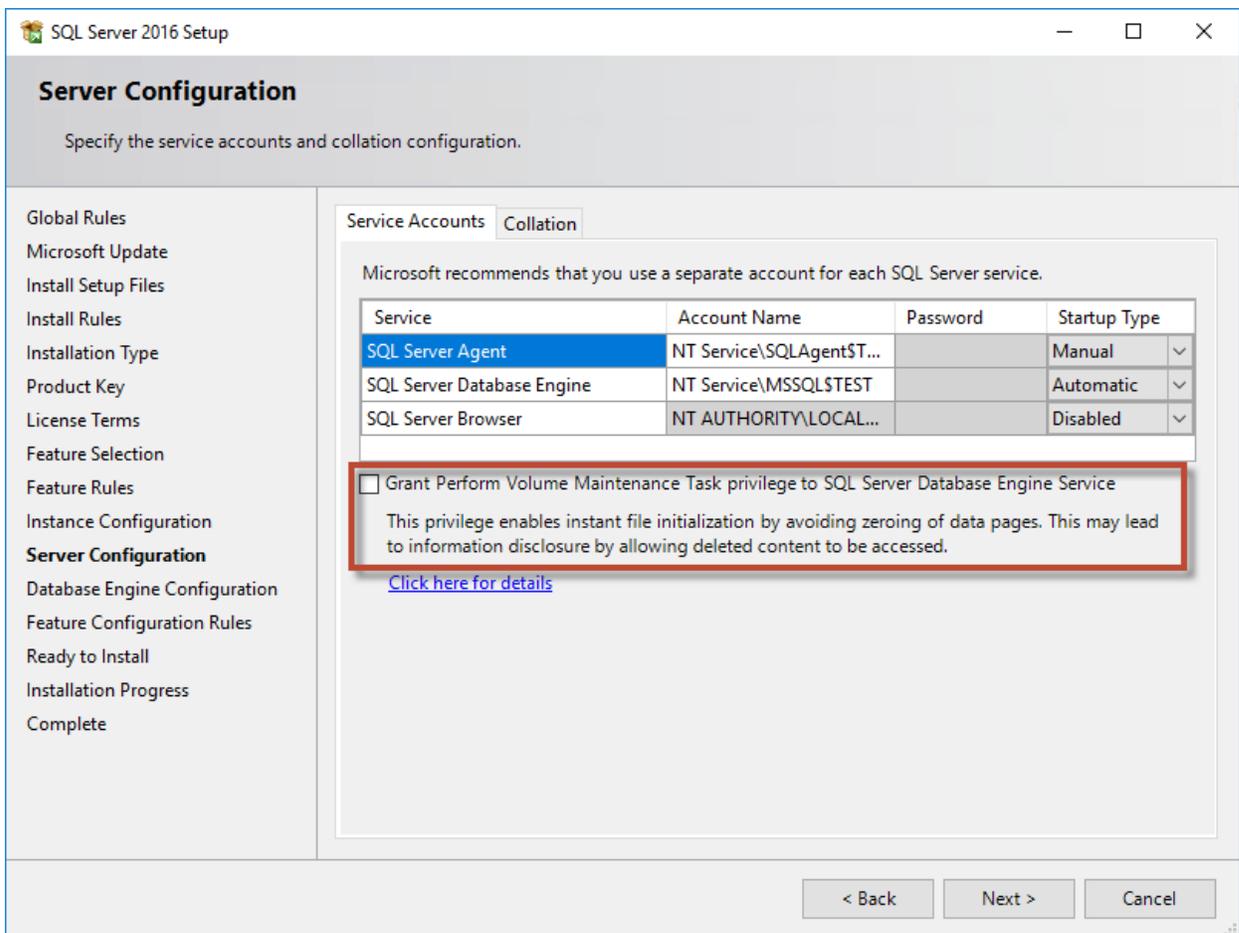
```
DBCC TRACEON(3004,3605,-1)
GO
CREATE DATABASE DelMe
GO
EXECUTE sp_readerrorlog
GO
DROP DATABASE DelMe
GO
DBCC TRACEOFF(3004,3605,-1)
GO
```

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	FixupLogTail(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。図6は、ボリュームメンテナンスタスクを実行する権限をSQL Serverデータベースエンジンサービスに付与するオプションを示しています。

図6) SQL Serverのインストール時にPerform volume maintenance task権限を付与するオプション



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、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することはできません。

データベースとストレージに関する考慮事項

NetAppのストレージ 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。

アグリゲート

アグリゲートは、NetAppストレージ構成のプライマリ ストレージ コンテナであり、データ ディスクとパリティ ディスクの両方で構成される1つ以上のRAIDグループを含みます。

NetAppでは、共有アグリゲートと専用アグリゲートを使用し、データ ファイルとトランザクション ログ ファイルを分離して、さまざまなI/Oワークロード特性評価テストを実施しています。このテストでは、1つの大規模なアグリゲートに複数のRAIDグループとスピンドルを追加することで、ストレージのパフォーマンスが最適化され、向上します。管理者は次の2つの理由から管理が楽になります。

- 1つの大きなアグリゲートで、すべてのスピンドルの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。

HA（高可用性）の場合は、SQL Server AlwaysOn 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ストレージ仮想マシン（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。

アグリゲートでは、ストレージのパフォーマンスを最適化するために、空きスペースを少なくとも10%確保することを推奨します。

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。1つのアグリゲートに多数のボリュームを作成でき、各ボリュームを拡張、縮小、またはアグリゲート間で移動できます。ユーザのダウンタイムは発生しません。

ボリュームの設計に関する考慮事項

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よってどのように変化するかを理解することが重要です。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。

- フレキシブル ボリュームを使用してSQL Serverデータベース ファイルを格納し、ホスト間でボリュームを共有しないようにします。
- 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。

- **SQL Server**データベースのI/Oプロファイルが、意思決定支援システムのワークロードなど、大部分がランジシーケンシャルリードで構成されている場合は、ボリュームで読み取りの再割り当てを有効にします。読み取りの再割り当ては、パフォーマンスを向上させるためにブロックを最適化します。
- **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Unicodeが有効になっていることを確認します。
- snapshot copy reserve 運用上の観点から監視しやすいように、ボリュームのNetApp値をゼロに設定します。
- ストレージSnapshot™ コピー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のボリュームまたはVMDK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、別の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、ほか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、優先度が高くありません。
- 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。このため、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、ログファイルの順次書き込みI/Oがデータファイルのランダム読み込み/書き込み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。
- 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。

LUN 数

tempdbファイル

ディスクの断片化を回避するために、tempdbファイルを事前にフルサイズに拡張することを推奨します。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。ラッチが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。これにより、クエリの実行時間が延長され、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C:\MSSQL\tempdb SQL Server 2012以降のマウントポイントに移動することで、tempdbを変更します。

```
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。

ホスト ログ ディレクトリ

SnapCenterはホスト ログ ディレクトリを使用して、トランザクション ログ バックアップ データを格納します。これはホストレベルです。SnapCenterで使用するSQL Serverホストごとに、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 × 日次ログ変更率%) × (Snapshotコピーの保持) ÷ (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。

ホスト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 → 設定プラグインで設定できます。

ホストログディレクトリに関するNetAppの推奨事項は次のとおりです。

- ホストログディレクトリが、バックアップSnapshotコピー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 データベースやシステム データベースを配置しないでください。
- 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。移行プロセスは中断を伴うため、移行の進行中にデータベースがオフラインにな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 クラスタ インスタンス (FCIS) には、次の条件が設定されている必要があります。
 - フェイルオーバークラスタイ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タイ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。

LUN上のデータベースに関するNetAppの推奨事項は次のとおりです。

- ユーザデータベースとホスト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テクノロジーで使用されるSnapshotコピーが保持ポリシーによって上書きされないようにします。
- 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をほかの.mdfファイルと共有していない場合にのみ有効です。
- 可能な場合は、SnapCenter Plug-in for Microsoft Windowsを使用してLUNを作成します。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。

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データベースの場合は、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。データの更新と要求を同時に行うアプリケーションの場合、ログファイルの書き込み処理は大量に行われ、データファイル（アプリケーションによって異なります）の読み取り/書き込み処理は大量に行われます。データの取得には、ログファイルは必要ありません。そのため、独自のドライブに配置されたデータファイルからデータの要求を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、データとログに別々のドライブを指定することをお勧めします。データベースの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する](#)」を参照してください。

Storage Efficiencyと管理性

Storage Efficiencyとは、消費するストレージスペースを最小限に抑えながら、システム全体のパフォーマンスにほとんど影響を与えずに、SQL Serverデータを格納、管理できる機能です。Storage Efficiencyは、データ重複排除にとどまらず、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。

次の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することで、ストレージ効率を実現し、コスト削減を実現します。これらのテクノロジーを組み合わせることで、削減効果が大きくなります。

Snapshotコピー

NetAppのSnapshotテクノロジーは、ONTAPアーキテクチャのWAFLE[®] 整合ポイントを作成してファイルシステム（ボリューム）またはLUNのポイントインタイムコピーを作成することで、高速で低コストのバックアップを実現します。NetApp SnapCenterは、SQL Server仮想デバイスインターフェイス（VDI）と統合されているため、本番SQL Serverデータベースのアプリケーションと整合性のあるSnapshotコピーを作成できます。本番データベースのダウンタイムは発生しません。

データが移動されないため、他のコピーアウトテクノロジーとは異なり、Snapshotコピーの作成時にパフォーマンスが低下することはありません。Snapshotコピーのコストは、ミラーコピーのようにバックアップごとに100%発生するわけではなく、ブロックレベルで変更された割合だけ発生します。Snapshotテクノロジーを使用すると、バックアップとリストアのためのストレージコストを大幅に削減でき、複数の効率的なデータ管理機能を利用できます。

データベースが同じボリューム上の複数のLUNを使用する場合、Snapshotコピーはボリュームベースであるため、これらのLUNのすべてのSnapshotコピーが同時に作成されます。特定の状況ではSnapCenter、SnapCenter検証プロセス中に書き込み可能なSnapshotコピーを使用して、SnapshotコピーからLUNをリストアし、一時的な読み取り/書き込みアクセスを可能にします。

シンプロビジョニング

シンプロビジョニングは、使用可能なストレージスペースの使用を最適化する方法です。この方法では、従来の方法のようにすべてのブロックを事前に割り当てるのではなく、データブロックをオンデマンドで割り当てます。この方法では、ほとんどすべての余白が排除されるため、利用率の低下を回避できます。

FlexVolは、NetAppのシンプロビジョニングを支えるテクノロジーであり、ONTAPの仮想化レイヤと考えることができます。LUNが作成されると、ボリュームから特定のデータブロックを、LUN自体またコピー専用にすることはありません。代わりに、データが実際に書き込まれるときにNetAppアグリゲートのブロックを割り当てます。この割り当て方法を使用すると、管理者は、ストレージシステムで物理的に使用可能なスペースよりも、接続されているサーバから見て多くのストレージスペースをプロビジョニングできます。

ストレージ消費を予測できない場合や変動が大きい場合は、急激な増加に対応できるように、ストレージのオーバーコミットメントのレベルを低くすることを推奨します。ストレージのコミットメントを100%（オーバーコミットメントなし）に制限し、トレンド分析機能を使用して、オーバーコミットメントが許容可能かどうかを判断することを検討してください。ストレージのオーバーコミットメントは、最小限のシステム停止でも許容できないミッションクリティカルなアプリケーション（SQL Serverなど）に対して慎重に検討し、管理する必要があります。このような場合は、ストレージの消費傾向を監視して、オーバーコミットメントの許容量を判断することを推奨します。

新しいストレージを調達するのに時間がかかる場合は、その時間を考慮してストレージのオーバーコミットメントしきい値を調整してください。管理者が新しいストレージを調達して導入する時間を確保できるように、オーバーコミットメントしきい値を余裕を持った値に設定する必要があります。Exchange環境でのシンプロビジョニング設定における潜在的なリスクは、それ以上データを書き込む十分なスペースがない場合に、LUNがオフラインになることです。ボリュームの自動拡張機能を軽減メカニズムとして使用して、シンプロビジョニングと高いストレージ使用率を安全に許可します。

スペースギャランティ

スペースギャランティにより、シンプロビジョニングが可能になります。スペースギャランティオプションは、ボリュームまたはLUNレベルで設定できます。ボリュームレベルのスペースギャランティがボリューム（デフォルト設定）に設定されている場合、FlexVolが必要とするスペースの量は、そのボリュームの作成時にアグリゲートから常に使用できます。

スペースギャランティを「none」に設定した場合は、ボリュームの作成時にアグリゲートからスペースがリザーブされません。スペースは、データをボリュームに書き込むとき、アグリゲートから確保されます。ギャランティを「none」に設定したボリュームで、スペースリザーブLUNに対して書き込み処理を行うと、その包含アグリゲートに十分な空き容量がない場合、処理は失敗します。

LUNリザーベーションは、LUNにボリューム内のスペースがあることを確認しますが、ギャランティを「none」に設定しても、ボリュームにアグリゲート内のスペースがあることは保証されません。ボリュームのスペースギャランティを「file」に設定した場合は、スペースリザーベーションを有効にしたLUNを完全に書き換えられるだけのスペースがアグリゲート内に確保されます。

SQL Server環境でシンプロビジョニングを使用して、スペース使用率を向上させ、スペースギャランティ機能を使用する場合のストレージ要件全体を削減することを推奨します。

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、最初にホスト上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。

フラクショナル リザーブ

フラクショナル リザーブは、ボリューム内のほかのすべてのスペースが使用されたあとに、LUNのSnapshotコピーの上書きに使用されるスペースを決定するボリューム オプションです。

NetAppストレージアプライアンスは、LUNシンプロビジョニングでさまざまな方法で設定でき、それぞれの方法に長所と短所があります。シンプロビジョニングされたボリュームとシンプロビジョニングされていないボリュームを、同じストレージシステム上または同じアグリゲート上に配置することもできます。次の2つのオプションは、SQL Serverでシンプロビジョニングを使用する場合のベストプラクティス構成とみなされます。

ボリュームギャランティをnoneに設定

表2の設定の利点は、アグリゲートの空きスペースが空きスペースの共有プールとして使用されることです。短所は、ボリューム同士の依存性が高いため、ボリュームごとに個別にシンプロビジョニングのレベルを調整することが難しいという点です。

表2の構成を使用する場合、ボリュームの合計サイズは、ホスト アグリゲートで使用可能な実際のストレージよりも大きくなります。通常、この設定を使用してボリュームのサイズを設定すると、ストレージ管理者は、アグリゲート内の使用済みスペースの管理と監視を行うだけでよくなります。このオプションは本番データをホストするスペースには影響しない一方で、バックアップ スペースは動的に変化するようになります。

表2) ボリュームギャランティがnoneに設定されていること

設定	Value
ボリューム ギャランティ	なし
LUNリザベーション	enabled
fractional_reserve	0%
snap_reserve	0%
自動削除	volume/oldest_first
autosize	オフ
try_first	snap_delete

自動拡張と自動削除の使用

表3の構成では、SQL Server環境のシンプロビジョニングのレベルを細かく調整できます。この設定では、ボリューム サイズにより、ボリューム内のLUNでのみ使用可能なスペースの量が定義（保証）されます。アグリゲートは、そのアグリゲートに含まれるすべてのボリュームで使用可能なスペースの共有ストレージプールの役割を果たします。

LUNまたはSnapshotコピーがボリュームで使用可能なスペースよりも多くのスペースを必要とする場合、ボリュームは自動的に拡張し、包含アグリゲートからより多くのスペースを取得します。さらに、LUNスペース リザベーション設定を無効にする利点は、LUNが必要としないスペースをSnapshotコピーで使用できることです。Snapshotコピーがスペースを占有するようになると、自動削除機能により削除されるため、LUN自体のスペースが足りなくなる危険もありません。

注：NetApp FlexClone® ボリュームの作成に使用されるSnapshotコピーは、自動削除オプションでは削除されません。

表3) 自動削除と自動拡張を使用したボリュームのセットアップ

設定	Value
ボリューム ギャランティ	ボリューム
LUNリザベーション	無効
fractional_reserve	0%

設定	Value
snap_reserve	0%
自動削除	volume/oldest_first
autosize	on
try_first	自動拡張

NetAppでは、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、一般的な構成のほとんどでは自動拡張を使用することを推奨しています。

NetApp FlexClone

NetApp FlexCloneテクノロジーを使用すると、デスティネーションFlexVolの読み書き可能なコピーをすばやく作成できます。データを追加でコピーする必要はもうありません。FlexCloneボリュームは、次のような場合に適しています。

- テストまたは開発が行われる場合
- 段階的な改善をロックすることで進行が行われた場合
- オリジナルの整合性を損なうことなく、変更可能な形式でデータを配布したい場合

FlexCloneを使用する一般的なシナリオは、ロールアップパッチやホットフィックスのインストール前にテストや開発を行う場合です。

FlexCloneテクノロジーは、プライマリストレージシステムとSnapMirrorデスティネーションの両方で、リソースを効果的に使用するために利用できます。また、FlexCloneを使用すると、SQL Server環境の運用を中断することなくディザスタリカバリのテストを行うこともできます。

SQL Serverでは、Snapshotに基づいてFlexCloneボリュームを作成するために、アプリケーションと整合性のあるSnapshotコピーが必要です。SnapCenterを使用して、Snapshotコピーを作成することを推奨します。

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

圧縮、コンパクション、重複排除

圧縮と重複排除の2つのStorage Efficiencyオプションを使用すると、所定の量の物理ストレージに格納できる論理データの量を増やすことができます。圧縮とは、簡単に言うと、データのパターンを検出してスペースが低減するようにエンコードする数学的なプロセスです。一方重複排除は、データ内で繰り返されるブロックを検出し、余計なコピーを取り除きます。どちらも結果にはそれほど違いはありませんが、仕組みが大きく異なるため管理の仕方も異なります。

ONTAP 9で導入されたインラインデータコンパクションにより、圧縮効率が向上します。アダプティブ圧縮では4KのWAFLブロックに8KのI/Oが格納されるため、実現する削減率は最大でも2:1です。二次圧縮のような圧縮手法は、8Kよりも大きいサイズのブロックを使用するので効率性に優れていますが、小さなブロックを上書きするデータには適しません。32KBのデータユニットを解凍して8K分を更新し、再度圧縮してからディスクに改めて書き込む処理では、オーバーヘッドが発生します。

圧縮

データベースの圧縮方法はいくつかあります。最近まで、データベースで十分なパフォーマンスを提供するには多数のスピンダルが必要なため、圧縮の効果は限られていました。許容可能なパフォーマンスでストレージアレイを構築すると、一般に必要以上の容量がアレイに提供されるという副作用がありました。この状況はソリッドステートストレージの普及で変わりました。優れたパフォーマンスを得るために、ドライブを過度にオーバープロビジョニングする必要がなくなったのです。

圧縮を使用しなくても、データベースをソリッドステートストレージプラットフォームに移行すれば（HDDとの混在構成かオールSSDかは問わない）、I/Oのサポートのためだけにドライブを購入する必要がなくなり、コストを大幅に削減できます。たとえば、NetAppは最近の大規模データベースプロジェクトのストレージ構成をいくつか対象に、NetApp Flash Cache™およびFlash Pool™のインテリジェントデータキャッシングまたはオールフラッシュアレイを使用して、ソリッドステートドライブ（SSD）を使用する場合と使用しない場合のコストを比較しました。

上述のように、ソリッドステートドライブ（SSD）ではIOPSが向上するため、ほぼすべてのケースでコストを削減できますが、さらに圧縮を使用すれば、ソリッドステートメディアの実効容量を増やして、削減効果をさらに高められます。

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。行圧縮はデータストレージ形式を変更します。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、行圧縮とその他の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](#)」を参照してください。

データ圧縮は、SQL Server 2008以降のEnterprise、Developer、およびEvaluation Editionで現在サポートされています。圧縮はデータベース自体から実行することも可能ですが、SQL Server環境ではこの方法がとられることは滅多にありません。

NetAppアダプティブ圧縮

アダプティブ圧縮は、SQL Serverワークロードで徹底的にテストされており、遅延がマイクロ秒単位で測定されるオールフラッシュ環境（デフォルトで有効になっている環境）でも、パフォーマンスへの影響はごくわずかです。最初のテストでは、一部のお客様から、圧縮によってパフォーマンスが向上したという報告もありました。これは、圧縮によって、データベースに使用できるSSDの容量が増えたことによります。

ONTAPは物理ブロックを4KB単位で管理するので、8KBのブロックを使用する一般的なSQL Serverデータベースの場合、最大で2:1の圧縮を実現できます。実際のユーザーデータを使った初期のテストではこれに近い圧縮率が観察されましたが、実際の結果は、格納されているデータのタイプによって異なります。

NetApp二次圧縮

二次圧縮では、使用するブロックのサイズがアダプティブ圧縮よりも大きく、32KBに固定されます。このためONTAPはデータを効率よく圧縮できますが、二次圧縮は主に、保存中のデータやシーケンシャルに書き込まれるデータなど、最大限の圧縮が必要なデータ向けに設計された機能です。

トランザクションログやバックアップファイルなどのデータには、二次圧縮を使用することを推奨します。これらのタイプのファイルはシーケンシャルに書き込まれ、更新されません。だからと言ってアダプティブ圧縮は勧められないということではありませんが、格納中のデータが大量にある場合は、アダプティブ圧縮よりも二次圧縮の方が削減効果が高まります。

データ量が膨大で、データファイルそのものが読み取り専用またはほとんど更新されない場合は、二次圧縮の使用を検討してください。32KBのブロックサイズを使用するデータファイルには、同じく32KBのブロックサイズを使用する二次圧縮を使用した方が、高い圧縮効果が得られます。ただし、圧縮するボリュームに、ブロックサイズが32KB以外のデータが格納されていないことを注意して確認する必要があります。また、二次圧縮は、頻繁に更新されないデータのみを使用してください。

インラインデータ コンパクション

インラインデータ コンパクションでは、論理的なWAFLブロックを物理的なWAFLブロックに格納することが可能です。たとえば、テキストブロックや部分的にフルブロックなど、圧縮率の高いデータを含むデータベースは、8KBから1KBに圧縮できます。コンパクションを使用しない場合、この1KBのデータによって、4KBブロック全体がその後も占有されます。

インラインデータ コンパクションを使用すれば、圧縮した1KBのデータを、他の圧縮データと一緒に1KBちょうどの物理スペースに格納できます。これは、圧縮技術ではなく、これは、ディスク上のスペースをより効率的に割り当てる方法であるため、検出できるほどのパフォーマンスへの影響はありません。

得られる削減効果の程度はさまざまです。圧縮済みのデータや暗号化データは、通常それ以上は圧縮することができないため、コンパクションによるメリットはありません。ブロック メタデータかせいぜいゼロ ブロックを含む場合、新しく初期化されたSQL Serverデータ ファイルは、最大80 : 1で圧縮されます。これによりもたらされる可能性ははかり知れません。どれくらいの削減効果を実現できるかを検証する一番の方法は、NetAppのスペース削減試算ツール (SSET) を使用することです。このツールはNetApp Field Portalから、またはNetAppの担当者経由で入手可能です。

重複排除

SQL Serverデータベース ファイルで重複排除を使用することは推奨されません。主に、このプロセスはほとんど効果がありません。SQL Serverページには、データベースにグローバルに固有のヘッダーと、ほぼ一意のトレーラが含まれています。重複排除によってスペースは1パーセント削減できますが、深刻なオーバーヘッドが発生するという犠牲が伴います。

多くの競合企業は、SQL Serverデータベースが何回も複製されていると仮定すれば、重複排除が可能だと主張しています。この点では、NetAppの重複排除も有効ですが、ONTAPにはNetApp FlexCloneというより優れたオプションがあります。結果はどちらも同じで、基盤の物理ブロックのほとんどを共有するSQL Serverデータベースのコピーが複数作成されます。FlexCloneを使用すれば、時間をかけてデータ ファイルを複製し、その後重複を排除するよりも、はるかに効率的です。FlexCloneでは、重複を排除するのではなく、二重になる部分が最初から作成されません。

同じデータ ファイルから作成されたコピーが複数存在するまれなケースについては、重複排除を使用できます。

NetAppでは、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、同じデータの複数のコピーがボリュームに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。

NetApp SnapMirror

NetApp SnapMirrorテクノロジーは、LANやWAN経由でデータをミラーリングまたは複製するための、高速で柔軟なエンタープライズ ソリューションを提供します。SnapMirrorテクノロジーは、最初のベース転送後に変更された4KBのデータ ブロックのみをデスティネーションに転送し、ネットワーク帯域幅の要件を大幅に削減します。

SnapMirrorは、設定された複製更新間隔に基づいて、非同期ボリュームレベルの複製を提供します。

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

- デスティネーションSVMは、ソースSVMがメンバーになっている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 ネームスペースと同じパスとディレクトリ構造にする必要があります。
- 整合性を保つために、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。ただし、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を更新してください。
- SQL Serverデータを含むボリュームをクラスタ内の異なるノードに分散し、すべてのクラスタ ノードが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。
- 可用性グループが使用するCIFS共有を、災害復旧のためにセカンダリ データセンターにミラーリングします。

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

NetAppクラウドデータサービスとSQL Server

ターゲットがオールクラウド、ハイブリッドクラウド、マルチクラウドのいずれであっても、NetAppクラウドデータサービスは、データ要件をアプリケーションレイヤでシームレスに実現することで、SQL Serverデータベースの導入や開発にかかる時間を短縮します。NetApp ONTAPテクノロジーに基づくネイティブのネットワークファイルシステムサービスをサポートする、可用性の高い解決策をクラウド内に作成できます。

SQL Server解決策およびNetAppクラウドデータサービスの詳細については、「[SQLデータベースでデータインテリジェンスを取得する](#)」を参照してください。

まとめ

SQL Serverユーザは、通常、SQL Serverへの投資回収率を高め、インフラを最適化してビジネス要件とIT要件をサポートするために、一連の大きな課題に直面しています。次のことが必要です。

- 新しいデータベースの導入や移行を迅速化し、これらの運用のリスクを軽減します。
- パフォーマンス、拡張性、可用性などのSLAをサポートするように、基盤となるストレージインフラが完全に最適化されていることを確認します。
- 既存のデータベースとインフラを統合してコストを削減
- 複雑さを軽減し、ITインフラを簡素化します。
- IT担当者の生産性を向上します。

こうした課題を克服するために、設計者、システム管理者、DBAは、実証済みのベストプラクティスとテクノロジーに基づいて、データベースとストレージインフラを導入しようとしています。

このドキュメントでは、Microsoft SQL Server環境の設計、最適化、拡張に関するNetAppの推奨事項について説明します。この推奨事項は、実装によって大きく異なる場合があります。クラスタ認識や仮想化などのオプションを使用すると、さらに多様になります。最適なソリューションは、環境の技術的な詳細と、プロジェクトの背景にあるビジネス要件の2つによって決まります。

このドキュメントでは、次の分野で一般的な推奨事項を示します。

- SQL Serverワークロード タイプ
- SQL Serverの設定
- データベース ストレージ レイアウト
- ストレージの効率化

NetApp SnapCenterソフトウェアとPlug-in for SQL ServerおよびPlug-in for Microsoft Windowsを併用すれば、SQL Serverデータベースをすばやく簡単に保護できます。これらの製品を使用すると、アプリケーションと整合性のあるバックアップ、自動クローニング、SQL Serverデータベース、インスタンス、可用性グループのリストアとリカバリを実行できます。

NetAppとパートナーのプロフェッショナル サービス エキスパートは、複雑なプロジェクトのサポートを提供しています。プロジェクト中はサポートが不要な場合も、NetAppが初めてのお客様には、プロフェッショナル サービスを利用して高度なアプローチの設計に役立てていただくことを強く推奨します。

詳細情報の入手方法

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

- SQLデータベースのデータインテリジェンスを実現
<https://cloud.netapp.com/solutions/sql-server-database>

- NetApp Interoperability Matrix Tool
<http://mysupport.netapp.com/NOW/products/interoperability/>
- NetAppの製品ドキュメント
docs.netapp.com
- TPC
<http://www.tpc.org/>
- HammerDB
<http://www.hammerdb.com/>
- SQL Server 2019 : 最新のデータプラットフォームを実現
<https://www.microsoft.com/en-us/sql-server/sql-server-2019-comparison>
- max worker threadsサーバ構成オプションの構成
<https://msdn.microsoft.com/en-us/library/ms190219.aspx>
- Microsoft SQL Server 2014環境に見るNetApp AFF8080 EXのパフォーマンスとサーバ統合
<https://fieldportal.netapp.com/content/248568?assetComponentId=248696>
- Page Compression Implementation
<https://msdn.microsoft.com/en-us/library/cc280464.aspx>
- ONTAP 9向けSnapMirrorの設定およびベストプラクティスガイド
<https://www.netapp.com/us/media/tr-4015.pdf>
- ONTAP 9.11.1向けSnapMirror Synchronous
<https://www.netapp.com/us/media/tr-4733.pdf>
- TR-4714 : 『Best Practice Guide for SQL Server Using SnapCenter』
<https://www.netapp.com/us/media/tr-4714.pdf>

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

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、および写真複製、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。

NetApp 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、NetApp によって「現状のまま」提供されています。NetApp 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。NetApp 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。

NetApp 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。NetApp 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、NetApp は責任を負いません。この製品の使用または購入は、NetApp 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り保護されている場合があります。

本書に含まれるデータは市販の製品および/またはサービス（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。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権を有し、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権については、DFARS 252.227-7015(b) 項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、<https://www.netapp.com/company/legal/trademarks/> 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。

TR-4590-0522-JP