



Microsoft SQL Server の場合

Enterprise applications

NetApp
February 10, 2026

目次

Microsoft SQL Server の場合	1
概要	1
Microsoft SQL Serverのワークロード	1
データベース設定	2
CPU構成	2
メモリ構成	5
共有インスタンスと専用インスタンス	8
tempdbファイル	9
AFF / FASシステムでのストレージ構成	10
概要	10
データベースファイルとファイルグループ	12
ストレージ効率	17
データ保護	22
ディザスタリカバリ	24
ASA R2システムでのストレージ構成	45
概要	45
データベースファイルとファイルグループ	47
データ保護	51
ディザスタリカバリ	52
データセキュリティ	68
Snapshot コピー	68
スナップショットの改ざん防止	68
SnapMirror レプリケーション	69
Storage Virtual Machine	69
カンリRBAC	69
多要素認証 (MFA)	69
API RBAC	69
マルチ管理者認証 (MAV)	70

Microsoft SQL Server の場合

概要

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
- Microsoft SQL Serverのアーキテクチャと管理
- 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

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データベースインスタンスがサポートするアプリケーションの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、次のように分類されます。

- OLTP：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、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、高いパフォーマンスが必要で、パフォーマンスの低下の影響を受けやすく、最大限の可用性が求められるSLAが用意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でのI/O構成は、通常、ランダムリードが75_{90%}、書き込みが25_{10%}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。データウェアハウスと呼ばれることもあります。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にとって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ほぼ100%の読み取りI/O構成であり、多くの場合、IOPSよりもI/Oスループットの方が重要です。

データベース設定

CPU構成

SQL Serverのパフォーマンスは、CPUとコアの構成にさまざまな依存関係があります。

ハイパースレッディング

ハイパースレッディングとは、同時マルチスレッド（SMT）の実装のことで、x86プロセッサで実行される計算の並列化を改善します。SMTはIntelプロセッサとAMDプロセッサの両方で使用できます。

ハイパースレッディングにより、論理CPUが物理CPUとしてオペレーティングシステムに認識されます。SQL Serverは、これらの追加CPUを認識し、物理的に存在するよりも多くのコアがあるかのように使用します。これにより、並列化を増やすことでパフォーマンスを大幅に向上させることができます。

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、[を参照してください "SQL Serverのエディション別のコンピューティング容量の制限"](#)。

コアとライセンス

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。

```

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）。また、各CPUに対してスケジューラfor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大幅に向上します。

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、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`12に設定する必要があります。`MAXDOP`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の場合、50から始めて、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`MAXDOP`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。

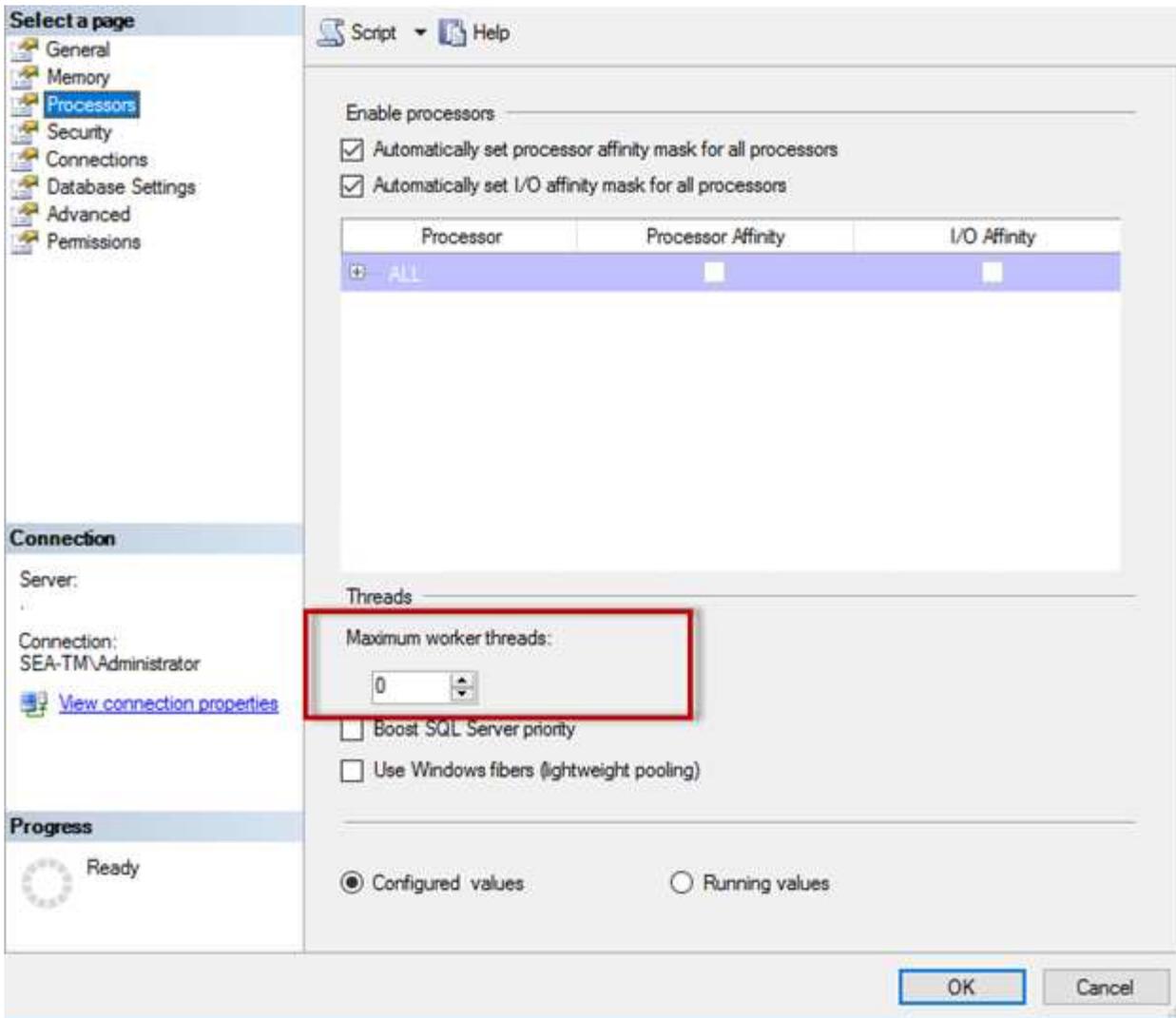
ワーカースレッドの最大数

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への同時接続数が数百に達すると、クエリごとに1つのスレッド構成で大量のシステムリソースが消費される可能性があります。このオプションを使用する`max worker threads`と、多数のクエリ要求をまとめて処理できるワーカースレッドのプールをSQL Serverで作成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、[を参照してください"max worker threadsサーバ構成オプションの設定"](#)。



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;  
GO
```

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に必要なSQL Serverのメモリ設定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、メモリに負荷がかかっても追加のメモリを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クラスターでは、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。

The screenshot shows the 'Server Memory Options' dialog box in SQL Server Enterprise Manager. The 'Memory' tab is selected in the left-hand 'Select a page' pane. The 'Server memory options' section is highlighted with a red box, showing 'Minimum server memory (in MB):' set to 0 and 'Maximum server memory (in MB):' set to 120832. Below this, the 'Other memory options' section shows 'Index creation memory (in KB, 0 = dynamic memory):' set to 0 and 'Minimum memory per query (in KB):' set to 1024. At the bottom, there are radio buttons for 'Configured values' (selected) and 'Running values', and 'OK' and 'Cancel' buttons.

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することもできます。

```
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、プロセッサバスの余分な負荷を回避するためのメモリアクセス最適化テクノロジーです。

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

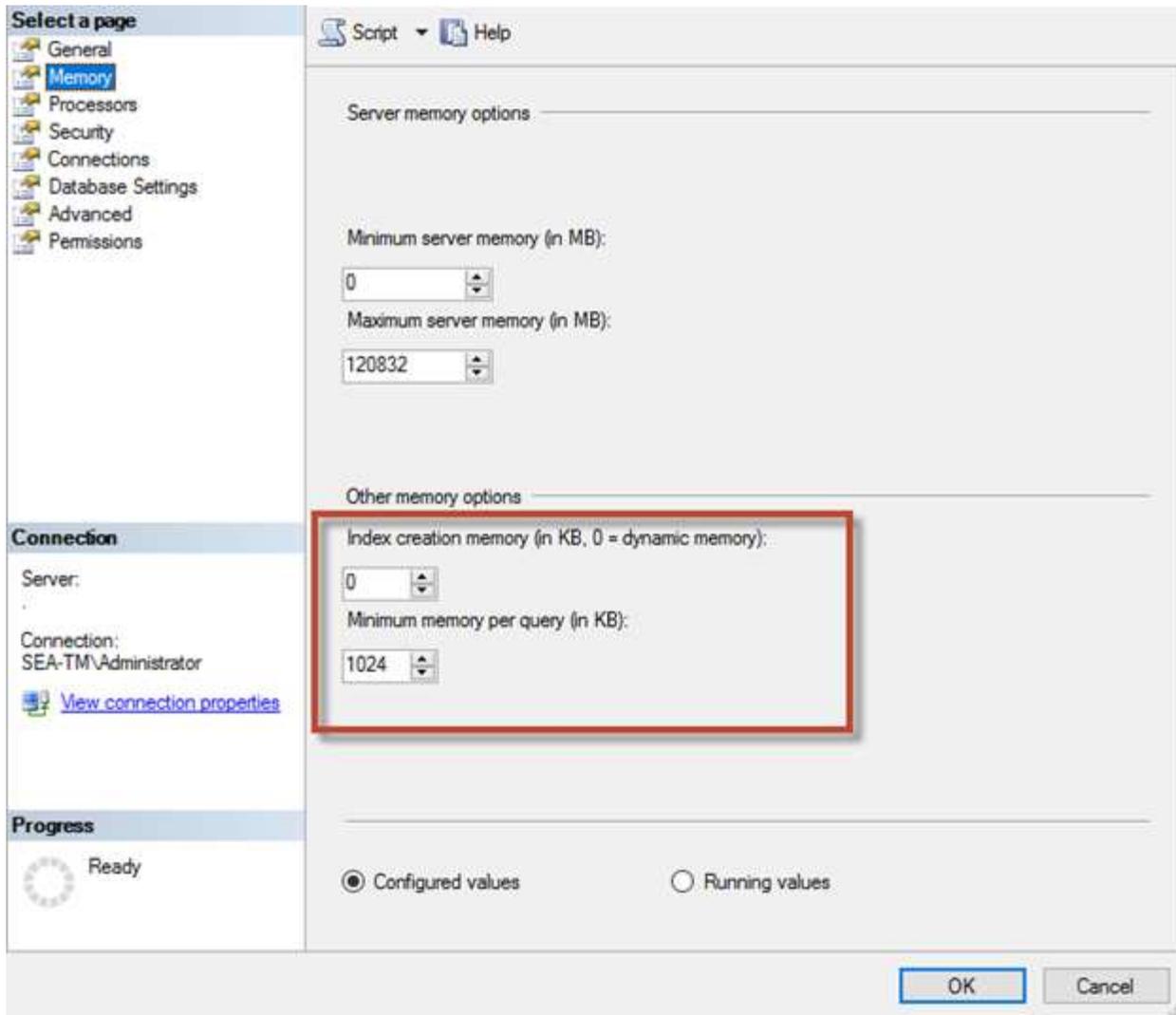
index create memoryオプションは、通常はデフォルトから変更する必要がないもう1つの詳細オプションです。

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。



共有インスタンスと専用インスタンス

SQL Serverは、サーバごとに1つのインスタンスとして設定することも、複数のインスタンスとして設定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設定は、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。

tempdbファイル

Tempdbデータベースは大量に利用できます。ユーザデータベースファイルをONTAPに最適に配置することに加えて、割り当ての競合を軽減するためには、tempdbデータファイルの配置も重要です。tempdbは別のディスクに配置し、ユーザデータファイルとは共有しないでください。

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、Global Allocation Map (GAM)、Shared Global Allocation Map (SGAM)、またはPage Free Space (PFS) ページで発生する可能性があります。ラッチによってこれらのページがメモリにロックされます。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
- tempdbデータファイルは同じサイズで作成する必要があります。

次のスクリプト例は、同じサイズのtempdbファイルを8つ作成し、tempdbを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。C:\MSSQL\tempdb

```
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。

AFF / FASシステムでのストレージ構成

概要

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が可能になります。

SQL Server on ONTAPソリューションを最適化するには、SQL ServerのI/Oパターンと特性を理解する必要があります。

あります。SQL Serverデータベース用のストレージレイアウトを適切に設計するには、SQL Serverのパフォーマンス要件を満たしながら、インフラ全体の管理性を最大限に高める必要があります。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

ボリュームはアグリゲート内に作成されて格納されます。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。

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。同じホストで複数のSQL Serverインスタンスを実行する場合は、ノードのボリューム制限に

近い場合を除き、ボリュームの共有は避け、データ管理を容易にするためにホストごとにインスタスごとに個別のボリュームを作成してください。

- 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。
- 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。
- (.mdf)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。また、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とデータファイルのランダムリード/ライトI/Oが分離され、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。
- データベースファイルとデータベース以外のファイル（フルテキスト検索関連ファイルなど）を同じLUNに混在させないでください。
- 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、データベースのファイルがLUNを他のファイルと共有していない`.mdf`場合にのみ有効`.mdf`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とネイティブ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。

データベースファイルとファイルグループ

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。

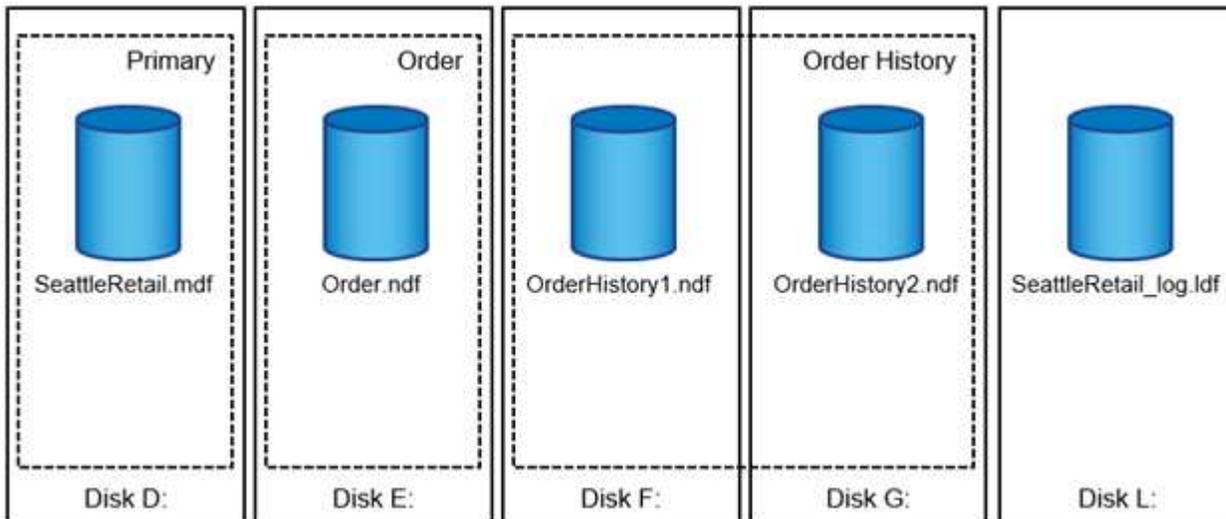
理論的には、SQL Server (64ビット) ではインスタンスあたり32、767個のデータベースと524、272TBのデータベースサイズがサポートされますが、通常のインストールでは複数のデータベースが使用されます。ただし、SQL Serverで処理できるデータベースの数は、負荷とハードウェアによって異なります。SQL Serverインスタンスでは、数十、数百、場合によっては数千の小規模データベースをホストしていることも珍しくありません。

データベースファイルとファイルグループ

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。与えられたワークロードをサポートするデータファイルの数は、アプリケーションに影響を与えることなく、I/O要件と想定容量をサポートするために必要に応じて変えることができます。データベースレイアウトのバリエーション

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。

データベースインスタンスファイルの初期化

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に、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。



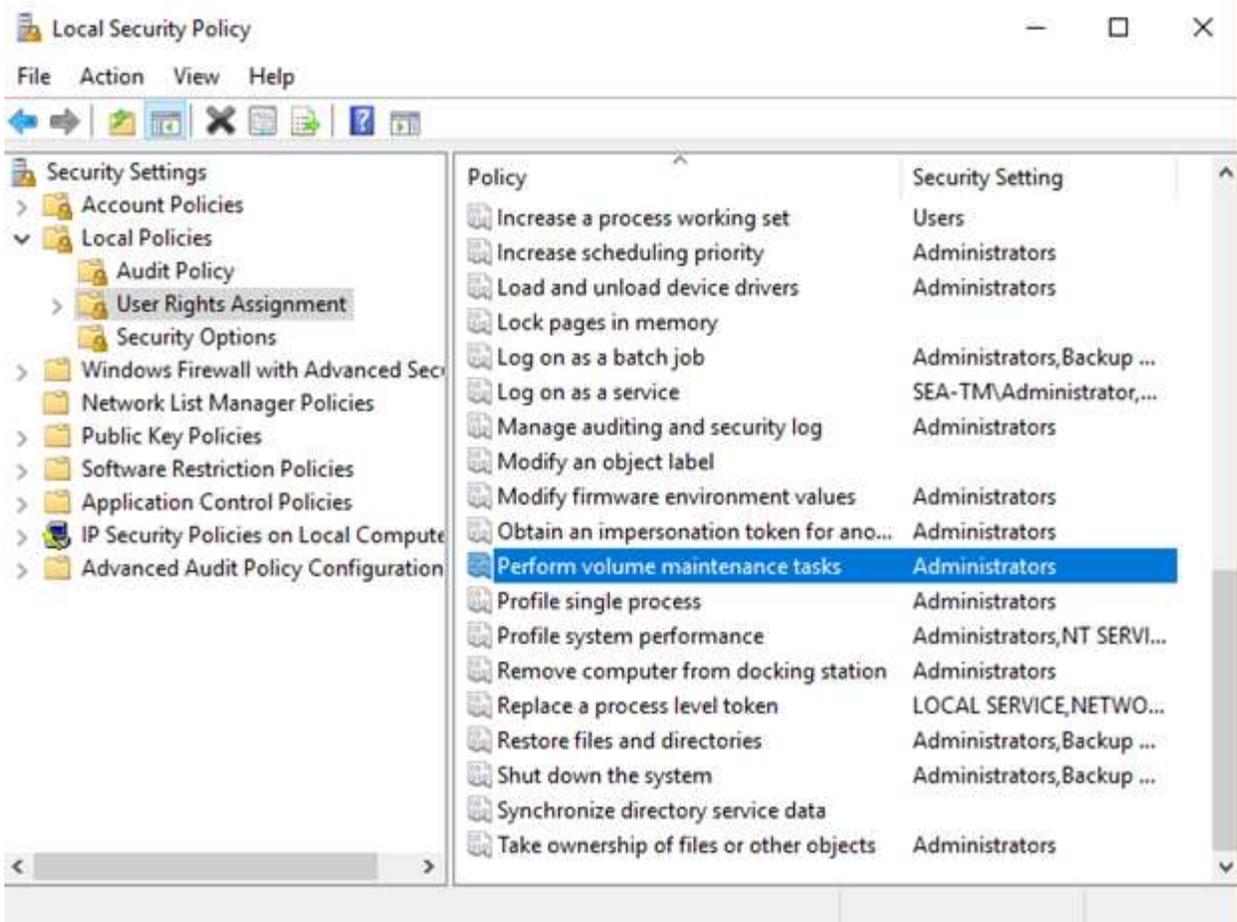
* NetAppでは、1つのファイルグループ内のすべてのファイルに同じ初期サイズと自動拡張パラメータを設定し、拡張サイズをパーセンテージではなくメガバイト単位で定義することを推奨しています*。これにより、Proportional Fill Algorithmは、データファイル間で書き込みアクティビティのバランスを均等に調整できます。

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。



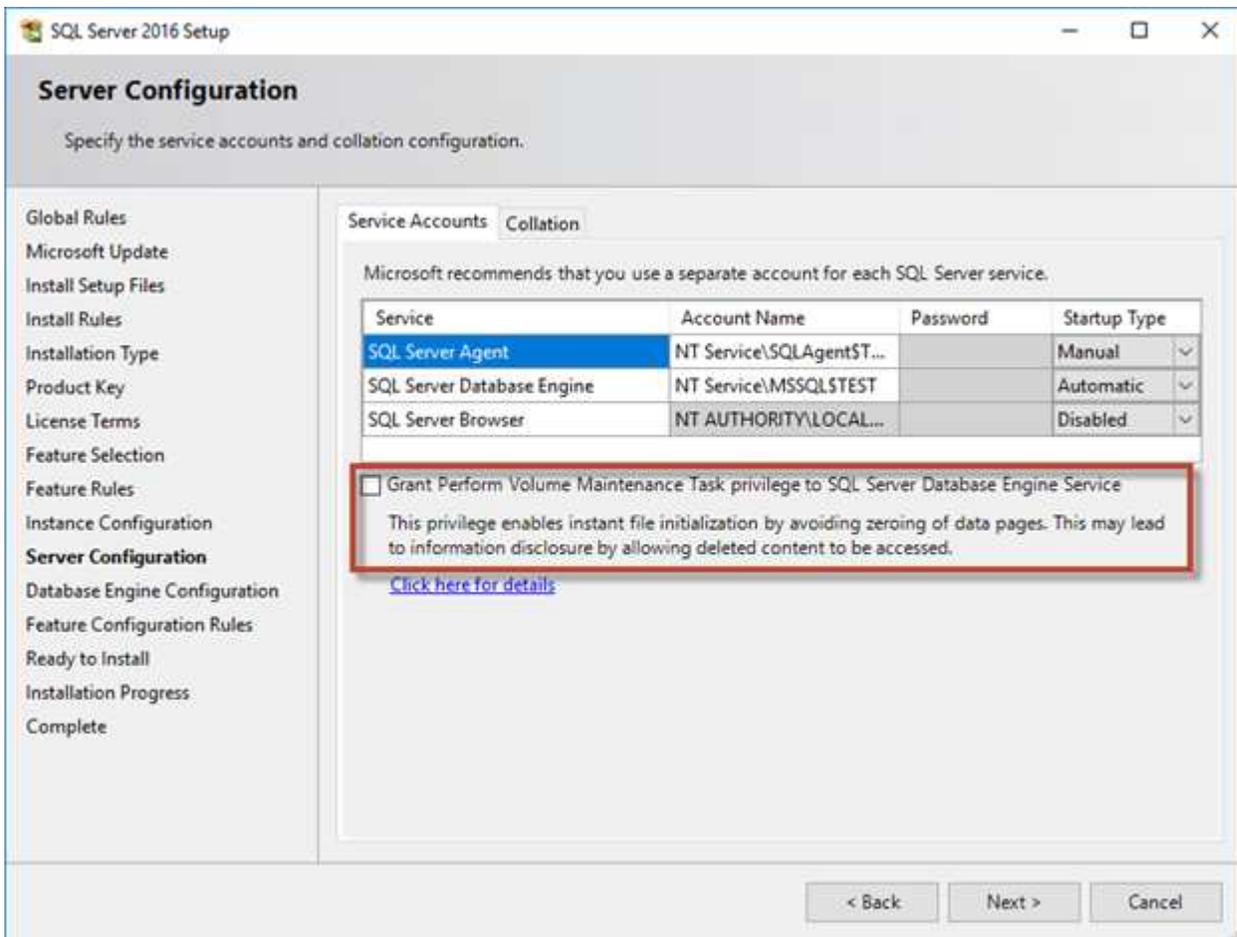
権限が有効になっているかどうかを確認するには、次の例のコードを使用します。このコードは、SQL Serverがエラーログに追加情報を書き込み、小さなデータベースを作成し、ログの内容を読み取るように強制する2つのトレースフラグを設定します。

```
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.

[Perform Volume Maintenance]タスクはSQL Server 2016では簡素化され、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。

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 사용되는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ボリュームに、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。



ストレージ効率

ONTAPのStorage Efficiencyは、パフォーマンスに影響を与えることなく消費するストレージスペースを最小限に抑えて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。

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。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

Temperature Sensitive Storage Efficiency (TSSE) は、ONTAP 9.8以降で使用できます。ブロックアクセスのヒートマップを使用して、アクセス頻度の低いブロックを特定し、より効率的に圧縮します。

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のヘッダーと、ほぼ一意の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。



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、コマンドを使用した解凍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。ボリュームは出力に表示されません volume efficiency show。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、設定がオーバーサイトされるとボリュームが予期せずスペース不足になり、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*：

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書ききれない場合は、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FAS上のボリュームで重複排除を有効にしないでください。

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。

データ保護

データベースのバックアップ戦略は、理論的な機能ではなく、特定されたビジネス要件に基づいて行う必要があります。ONTAPのSnapshotテクノロジーとMicrosoft SQL Server APIを組み合わせることで、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。

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"](#)。



* 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。前述の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。

SnapCenterを使用したSQL Server可用性グループ

SnapCenterでは、Windowsフェイルオーバークラスターで構成されたSQL Server可用性グループデータベースのバックアップがサポートされます。

SnapCenter Plugin for Microsoft SQL Serverは、Windowsサーバフェイルオーバークラスターのすべてのノードにインストールする必要があります。SnapCenterプラグインをセットアップするための前提条件と手順を参照してください["ドキュメント"](#)。

SnapCenterでは、Windowsホスト内のすべてのデータベース、インスタンス、および可用性グループが検出され、SnapCenterのリソースページにリソースが列挙されます。

Always-On可用性グループのデータベースの保護

可用性グループ内のデータベースは、複数の方法で保護できます。

- データベースレベルのバックアップ：データベースリソースページの可用性データベースを選択し、フル/ログバックアップを含むポリシーを追加し、バックアップをスケジュールします。SnapCenterは、プライマリレプリカであるかセカンダリレプリカであるかに関係なく、データベースのロールに関係なくバックアップを実行します。リソースグループにデータベースを追加して保護を設定することもできます。
- インスタンスレベルのバックアップ：インスタンスを選択すると、選択したポリシーに基づいてインスタンスで実行されているすべてのデータベースが保護されます。プライマリレプリカまたはセカンダリレプリカとして実行されている可用性データベースを含め、すべてのデータベースがSnapCenterを使用してバックアップされます。リソースグループにインスタンスを追加して保護を設定することもできます。
- 可用性グループレベルのバックアップ：ポリシーの設定時に、SnapCenterには可用性グループレベルのバ

ックアップの詳細オプションがあります。ポリシーの可用性グループ設定を使用すると、バックアップのレプリカ設定を選択できます。プライマリレプリカ、セカンダリレプリカ、またはそれらのすべてを選択できます。デフォルトのオプションは、SQL Server可用性グループ構成のバックアップレプリカセットに基づいています。

SnapCenterポリシーの可用性グループ設定は、可用性グループレベルのバックアップが可用性グループデータベースの保護に使用されている場合にのみ適用され、データベースレベルまたはインスタンスレベルのバックアップには適用されません。



* NetAppでは、可用性レベルのバックアップを使用して、NetApp ONTAPストレージで実行されているすべてのレプリカをバックアップすることを推奨しています。

SnapCenterでのログバックアップの設定

可用性グループをスタンドアロンSQL Serverセットアップでセットアップする場合は、Windowsサーバフェイルオーバークラスターの各ノードに専用ディスクをマウントする必要があります。トランザクションログバックアップを保存するログディレクトリを設定するには、専用ディスクを使用する必要があります。

SQL Serverフェイルオーバークラスターで可用性グループが設定されている場合は、SQL Serverフェイルオーバークラスターインスタンス上にホストログディレクトリへのクラスターディスクを作成する必要があります。

SnapCenterを使用した可用性グループセットアップでのデータベースのリストア

- SnapCenterには再シードオプションがあり、セカンダリレプリカにある最新のSnapshotからデータベースを自動的にリカバリできます。再シード処理を実行すると、データベースバックアップが自動的にリストアされて可用性グループに追加されます。
- 可用性グループ内のレプリカデータベースをリストアするもう1つの方法は、可用性グループを解除し、完全なフルリストアとログリストアを実行することです。SnapCenterを使用してデータベースをノーリカバリモードでリストアし、SQL Server Management StudioまたはT-SQLを使用してデータベースを可用性グループに戻します。
- データのサブセットだけをリカバリするには、SnapCenterのクローニング機能を使用してデータベースのクローンコピーを作成します。データベースのコピーは、SnapCenterを使用して数分以内に作成され、SQL Serverのネイティブツールを使用してプライマリレプリカにデータをエクスポートします。

RTOとRPOの要件を満たすようにデータベースストレージレイアウトを設定するためのベストプラクティスについては、[を参照してください"TR-4714 : NetApp SnapCenterを使用したMicrosoft SQL Serverのベストプラクティス"](#)。



SnapCenterでは、分散可用性グループと含まれる可用性グループはサポートされません。

ディザスタリカバリ

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には、データ保護とAlways-Onを統合するためのオプションが用意されています。ただし、ONTAPレプリケーションテ

クノロジを検討する必要がある場合もあります。3つの基本的なオプションがあります。

SnapMirror

SnapMirrorテクノロジーは、LANおよびWAN経由でデータを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。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。同期モードまたは非同期モードのいずれかで設定できます。

NetApp MetroClusterとSnapMirrorのアクティブな同期

多くのお客様にとって、DRに必要なのはデータのリモートコピーだけではなく、そのデータを迅速に活用できることです。NetAppは、このニーズに対応する2つのテクノロジーを提供します。MetroClusterとSnapMirrorのアクティブ同期です。

MetroClusterとは、低レベルの同期ミラーリングストレージと多数の追加機能を含むハードウェア構成のONTAPのことです。MetroClusterなどの統合ソリューションは、今日の複雑なスケールアウトデータベース、アプリケーション、仮想化インフラストラクチャを簡素化します。複数の外部データ保護製品や戦略を、1つのシンプルな中央集中型ストレージアレイに置き換えます。また、単一のクラスタストレージシステム内に、バックアップ、リカバリ、ディザスタリカバリ、高可用性（HA）が統合されています。

SnapMirrorアクティブ同期はSnapMirror Synchronousに基づいています。MetroClusterでは、各ONTAPコントローラがドライブデータをリモートサイトにレプリケートします。SnapMirrorアクティブ同期を使用すると、基本的には2つの異なるONTAPシステムでLUNデータの独立したコピーを維持しながら、このLUNの単一インスタンスを提供できます。ホストの観点からは、単一のLUNエンティティです。

SM-ASとMCCの比較

SM-ASとMetroClusterは全体的な機能が似ていますが、RPO=0レプリケーションの実装方法と管理方法には重要な違いがあります。SnapMirrorの非同期および同期はDR計画の一部としても使用できますが、HAレプリケーションテクノロジーとしては設計されていません。

- MetroCluster構成は、複数のサイトにノードが分散された統合クラスタのようなものです。SM-ASは、同期的にレプリケートされるRPO=0のLUNにサービスを提供する独立した2つのクラスタのように動作します。
- MetroCluster構成のデータには、常に1つの特定のサイトからしかアクセスできません。データの2つ目のコピーは反対側のサイトに存在しますが、データはパッシブです。ストレージシステムのフェイルオーバーがないとアクセスできません。
- MetroClusterとSM-ASによるミラーリングは、さまざまなレベルで実行されます。MetroClusterミラーリングはRAIDレイヤで実行されます。下位レベルのデータは、SyncMirrorを使用してミラーリングされた形式で格納されます。ミラーリングは、LUN、ボリューム、プロトコルの各レイヤでは実質的に使用されません。
- 一方、SM-ASミラーリングはプロトコルレイヤで行われます。2つのクラスタは、全体的に独立したクラスタです。データの2つのコピーが同期されると、2つのクラスタは書き込みをミラーリングするだけで済みます。一方のクラスタで書き込みが発生すると、もう一方のクラスタにレプリケートされます。書き込みの確認応答がホストに送信されるのは、両方のサイトで書き込みが完了した場合だけです。このプロトコルスプリット動作以外では、2つのクラスタは通常のONTAPクラスタです。
- MetroClusterの主な役割は大規模なレプリケーションです。RPO=0でRTOがほぼゼロのアレイ全体をレプリケートできます。フェイルオーバーが1つしかなく、容量とIOPSの点で非常に適切に拡張できるため、フェイルオーバープロセスが簡易化されます。
- SM-ASの主なユースケースの1つに、きめ細かなレプリケーションがあります。すべてのデータを1つのユ

ニットとしてレプリケートしたくない場合や、特定のワークロードを選択的にフェイルオーバーできる必要がある場合があります。

- SM-ASのもう1つの主なユースケースは、アクティブ/アクティブ処理です。アクティブ/アクティブ処理では、データの完全に使用可能なコピーを、同じパフォーマンス特性を持つ2つの異なるクラスタに配置し、必要に応じてSANをサイト間で拡張する必要がありません。アプリケーションを両方のサイトで実行しておくことで、フェイルオーバー処理中の全体的なRTOを短縮できます。

SnapMirror

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

- SMB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。SMBを使用する場合は、デスティネーションNAS名前スペースをソース名前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。
- 高速データリカバリのニーズが高い場合は同期レプリケーションを使用し、RPOの柔軟性を確保するために非同期ソリューションを使用します。

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』](#)"。

MetroCluster

アーキテクチャ

MetroCluster環境でMicrosoft SQL Serverを導入するには、MetroClusterシステムの物理設計について説明する必要があります。

MetroClusterは、別々の場所または障害ドメインにある2つのONTAPクラスタ間でデータと設定を同期的にミラーリングします。MetroClusterは、次の2つの目的を自動的に管理することで、アプリケーションに継続的可用性を提供します。

- クラスタに書き込まれたデータを同期的にミラーリングすることで、Recovery Point Objective（RPO；目標復旧時点）がゼロになります。
- 2番目のサイトで構成をミラーリングし、データへのアクセスを自動化することで、目標復旧時間（RTO）をほぼゼロに抑えます。

MetroClusterは、2つのサイトにある2つの独立したクラスタ間でデータと構成を自動的にミラーリングすることで、シンプルな作業を実現します。1つのクラスタ内でストレージがプロビジョニングされると、2つ目のサイトの2つ目のクラスタに自動的にミラーリングされます。NetApp SyncMirror®は、すべてのデータの完全

なコピーをゼロRPOで提供します。つまり、1つのサイトのワークロードをいつでも反対側のサイトに切り替えて、データを損失することなくデータの提供を継続できます。MetroClusterは、2番目のサイトでNASおよびSANでプロビジョニングされたデータへのアクセスを提供するスイッチオーバープロセスを管理します。検証済みソリューションとしてのMetroClusterの設計には、プロトコルのタイムアウト期間内またはそれよりも早く（通常は120秒未満）スイッチオーバーを実行できるようにサイジングと設定が含まれています。その結果、RPOがほぼゼロになり、アプリケーションは障害を発生させることなくデータへのアクセスを継続できます。MetroClusterには、バックエンドストレージファブリックで定義されたいくつかのバリエーションがあります。

MetroClusterは3種類の構成で使用できます。

- IPセツソクノHAペア
- FCセツソクノHAペア
- シングルコントローラ、FC接続



「接続」という用語は、サイト間レプリケーションに使用されるクラスタ接続を指します。ホストプロトコルを指しているわけではありません。MetroCluster構成では、クラスタ間通信に使用される接続の種類に関係なく、すべてのホスト側プロトコルが通常どおりサポートされます。

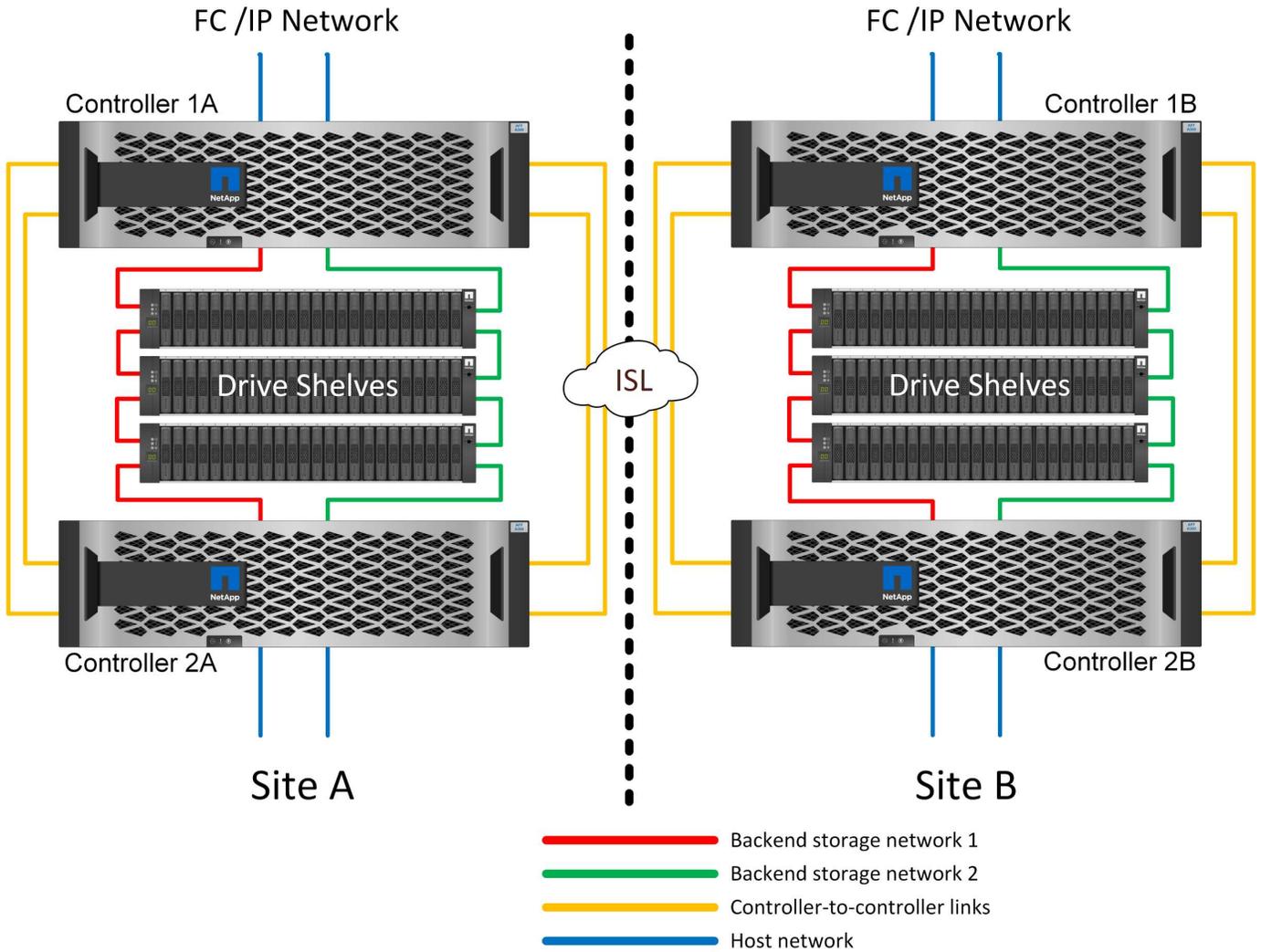
MetroCluster IP の略

HAペアMetroCluster IP構成では、サイトごとに2ノードまたは4ノードを使用します。この設定オプションを使用すると、2ノードオプションに比べて複雑さとコストが増加しますが、サイト内の冗長性という重要なメリットがあります。単純なコントローラ障害では、WAN経由のデータアクセスは必要ありません。データアクセスは、代替ローカルコントローラを介してローカルのままです。

ほとんどのお客様は、インフラストラクチャの要件がシンプルであるため、IP接続を選択しています。これまでは、ダークファイバやFCスイッチを使用した場合、サイト間での高速接続のプロビジョニングは一般的に容易でしたが、今日では、高速で低レイテンシのIP回線がより容易に利用可能になっています。

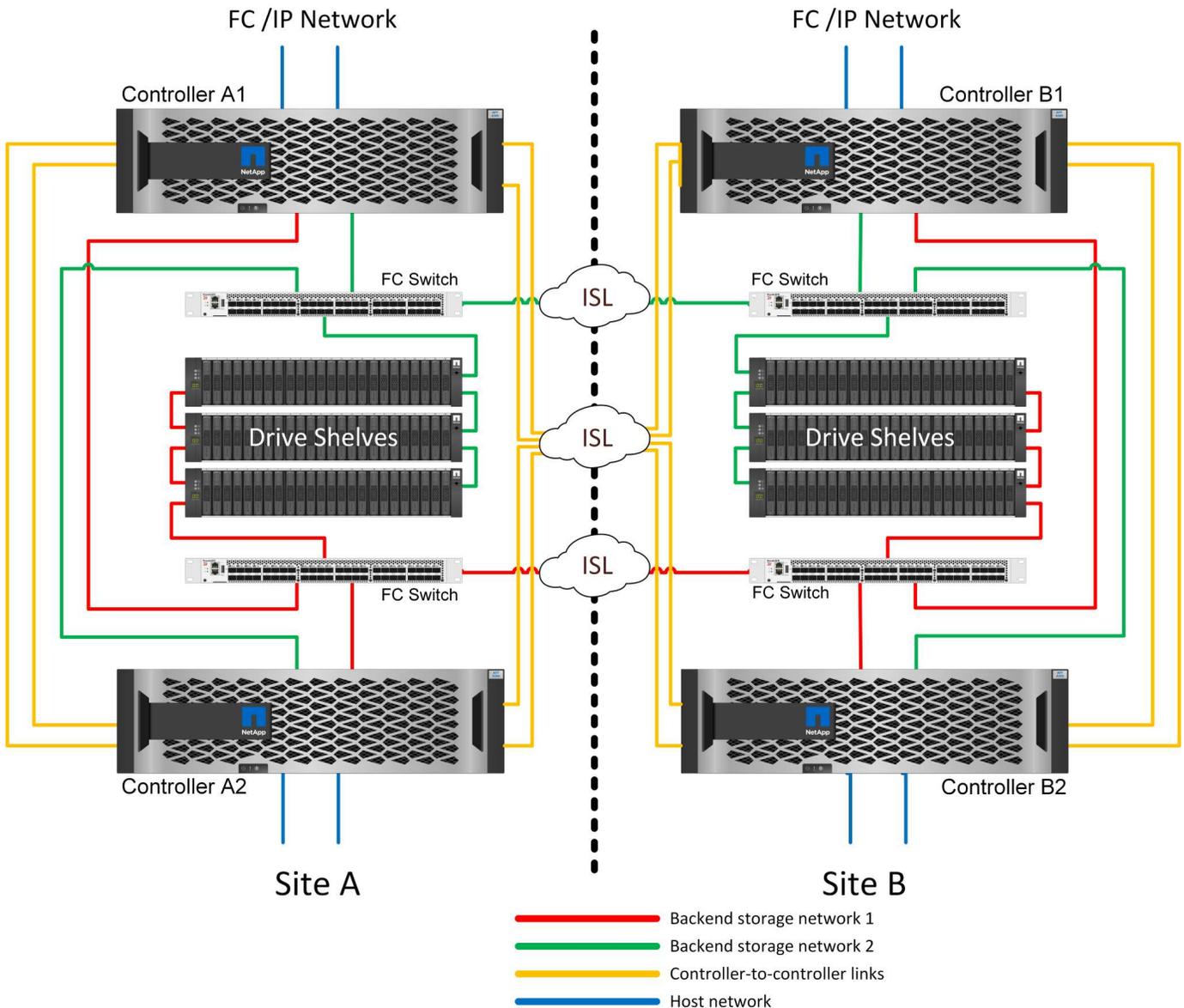
サイト間接続はコントローラのみであるため、アーキテクチャもシンプルです。FC SAN接続MetroClusterでは、コントローラが反対側サイトのドライブに直接書き込むため、追加のSAN接続、スイッチ、およびブリッジが必要になります。一方、IP構成のコントローラは、コントローラを介して反対側のドライブに書き込みます。

追加情報については、ONTAPの公式ドキュメントを参照してください。"[MetroCluster IP 解決策のアーキテクチャと設計](#)"。



HAペアFC SAN接続MetroCluster

HAペアMetroCluster FC構成では、サイトごとに2ノードまたは4ノードを使用します。この設定オプションを使用すると、2ノードオプションに比べて複雑さとコストが増加しますが、サイト内の冗長性という重要なメリットがあります。単純なコントローラ障害では、WAN経由のデータアクセスは必要ありません。データアクセスは、代替ローカルコントローラを介してローカルのままです。

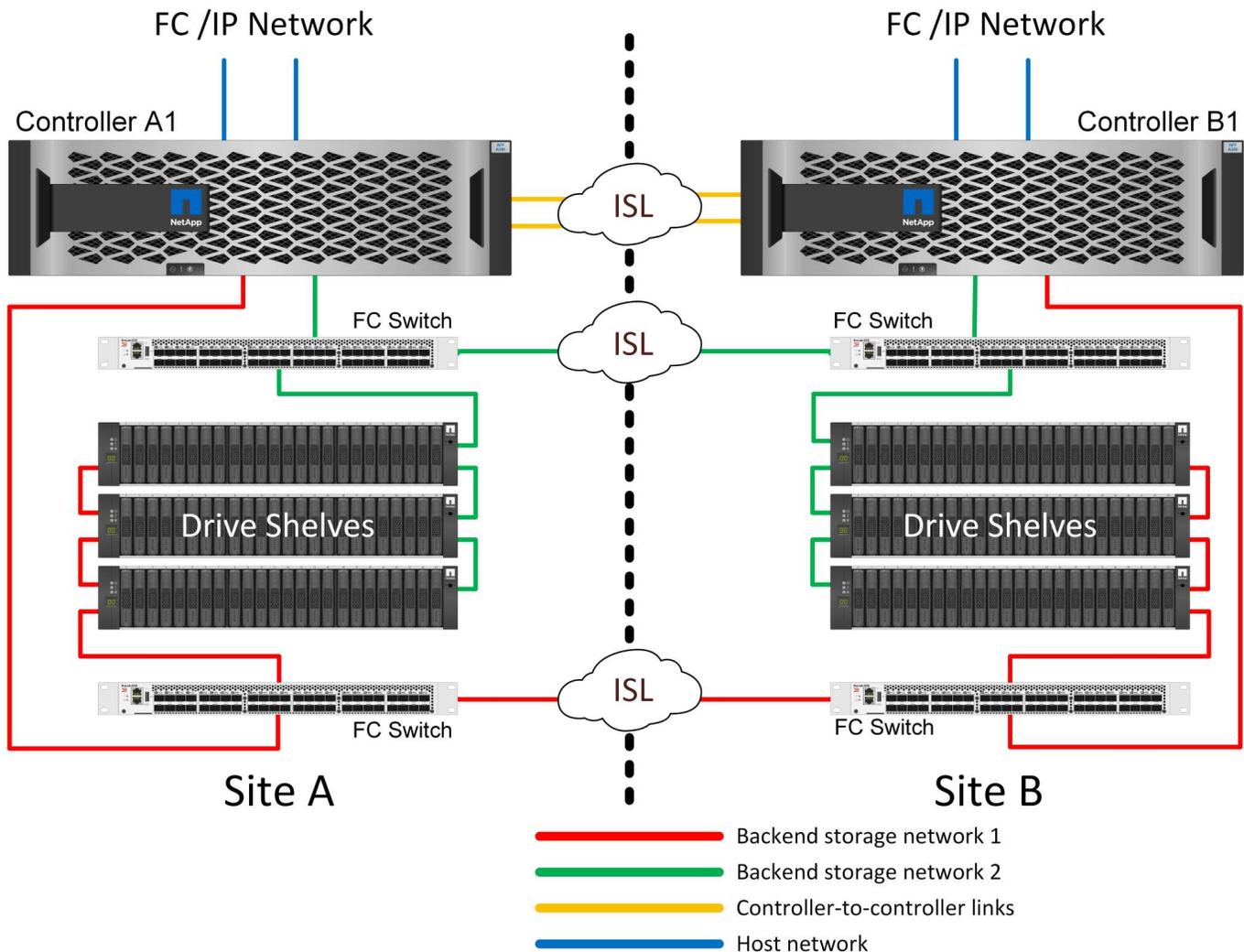


一部のマルチサイトインフラは、アクティブ/アクティブ運用向けに設計されたものではなく、プライマリサイトやディザスタリカバリサイトとして使用されます。この場合、一般にHAペアMetroClusterオプションが推奨される理由は次のとおりです。

- 2ノードMetroClusterクラスタはHAシステムですが、コントローラに予期しない障害が発生した場合や計画的メンテナンスを行う場合は、反対側のサイトでデータサービスをオンラインにする必要があります。サイト間のネットワーク接続が必要な帯域幅をサポートできない場合は、パフォーマンスが低下します。唯一の選択肢は、さまざまなホストOSと関連サービスを代替サイトにフェイルオーバーすることです。HAペアMetroClusterクラスタでは、コントローラが停止すると同じサイト内で単純なフェイルオーバーが発生するため、この問題は解消されます。
- 一部のネットワークポロジは、サイト間アクセス用に設計されていませんが、異なるサブネットまたは分離されたFC SANを使用します。この場合、代替コントローラが反対側のサイトのサーバにデータを提供できないため、2ノードMetroClusterクラスタはHAシステムとして機能しなくなります。完全な冗長性を実現するには、HAペアMetroClusterオプションが必要です。
- 2サイトインフラを単一の高可用性インフラとみなす場合は、2ノードMetroCluster構成が適しています。ただし、サイト障害後もシステムが長時間機能しなければならない場合は、HAペアが推奨されます。HAペアは、単一サイト内でHAを提供し続けるためです。

2ノードFC SAN接続MetroCluster

2ノードMetroCluster構成では、サイトごとに1つのノードのみが使用されます。設定とメンテナンスが必要なコンポーネントが少ないため、HAペアオプションよりもシンプルな設計になっています。また、ケーブル配線やFCスイッチの点でインフラストラクチャの必要性も軽減されています。最後に、コストを削減します。



この設計の明らかな影響は、1つのサイトでコントローラに障害が発生した場合、反対側のサイトからデータを利用できることです。この制限は必ずしも問題ではありません。多くの企業は、本質的に単一のインフラとして機能する、拡張された高速で低レイテンシのネットワークを使用したマルチサイトデータセンター運用を行っています。このような場合は、2ノードバージョンのMetroClusterが推奨されます。2ノードシステムは現在、複数のサービスプロバイダでペタバイト規模で使用されています。

MetroClusterの耐障害性機能

MetroCluster 解決策には単一点障害（Single Point of Failure）はありません。

- 各コントローラに、ローカルサイトのドライブシェルフへの独立したパスが2つあります。
- 各コントローラに、リモートサイトのドライブシェルフへの独立したパスが2つあります。
- 各コントローラには、反対側のサイトのコントローラへの独立したパスが2つあります。
- HAペア構成では、各コントローラからローカルパートナーへのパスが2つあります。

つまり、構成内のコンポーネントを1つでも削除しても、MetroClusterのデータ提供機能を損なうことはありません。2つのオプションの耐障害性の違いは、サイト障害後もHAペアバージョンが全体的なHAストレージシステムになる点だけです。

SyncMirror

MetroClusterを使用したSQL Serverの保護は、最大パフォーマンスのスケールアウト同期ミラーリングテクノロジーを提供するSyncMirrorに基づいています。

SyncMirrorによるデータ保護

最も簡単な意味では、同期レプリケーションとは、変更がミラーされたストレージの両側に対して確認応答される前に行われなければならないことを意味します。たとえば、データベースがログを書き込んでいる場合やVMwareゲストにパッチを適用している場合は、書き込みが失われることはありません。プロトコルレベルでは、両方のサイトの不揮発性メディアにコミットされるまで、ストレージシステムは書き込みを確認応答しないください。その場合にのみ、データ損失のリスクなしに作業を安全に進めることができます。

同期レプリケーションテクノロジーの使用は、同期レプリケーション解決策を設計および管理するための最初のステップです。最も重要な考慮事項は、計画的および計画外のさまざまな障害シナリオで何が発生するかを理解することです。すべての同期レプリケーションソリューションが同じ機能を提供するわけではありません。Recovery Point Objective (RPO；目標復旧時点) がゼロ (つまりデータ損失ゼロ) の解決策が必要な場合は、すべての障害シナリオを考慮する必要があります。特に、サイト間の接続が失われてレプリケーションが不可能になった場合、どのような結果が予想されますか。

SyncMirrorデータの可用性

MetroClusterレプリケーションは、同期モードに効率的に切り替えられるように設計されたNetApp SyncMirrorテクノロジーに基づいています。この機能は、同期レプリケーションを必要とする一方で、データサービスに高可用性も必要とするお客様の要件を満たします。たとえば、リモートサイトへの接続が切断されている場合は、通常、ストレージシステムをレプリケートされていない状態で運用し続けることを推奨します。

多くの同期レプリケーションソリューションは、同期モードでしか動作できません。このタイプのall-or-nothingレプリケーションは、Dominoモードと呼ばれることがあります。このようなストレージシステムでは、データのローカルコピーとリモートコピーが非同期になるのではなく、データの提供が停止します。レプリケーションが強制的に解除された場合、再同期には非常に時間がかかり、ミラーリングの再確立中にデータが完全に失われる可能性があります。

リモートサイトに到達できない場合にSyncMirrorを同期モードからシームレスに切り替えることができるだけでなく、接続がリストアされたときにRPO=0状態に迅速に再同期することもできます。再同期中にリモートサイトにある古いデータコピーを使用可能な状態で保持することもできるため、データのローカルコピーとリモートコピーを常に維持できます。

Dominoモードが必要な場合、NetAppはSnapMirror Synchronous (SM-S) を提供します。Oracle DataGuardやSQL Server Always On可用性グループなど、アプリケーションレベルのオプションも用意されています。オプションとして、OSレベルのディスクミラーリングを使用できます。追加情報とオプションについては、担当のNetAppまたはパートナーアカウントチームにお問い合わせください。

SQL ServerとMetroCluster

SQL ServerデータベースをゼロRPOで保護する方法の1つに、MetroClusterがあります。MetroClusterは、シンプルでハイパフォーマンスなRPO=0レプリケーションテクノロジーで、インフラ全体をサイト間で簡単にレプリケートできます。

SQL Serverは、1つのMetroClusterシステムで数千のデータベースまでスケールアップできます。SQL Serverのスタンドアロンインスタンスやフェイルオーバークラスティンスタンスが存在する場合もあります。MetroClusterシステムでは、データベース管理のベストプラクティスを追加したり、ベストプラクティスを変更したりする必要はありません。

MetroClusterの完全な説明は、このドキュメントの範囲外ですが、原則は簡単です。MetroClusterは、迅速なフェイルオーバーを備えたRPO=0レプリケーションソリューションを提供します。この基盤の上に何を構築するかは、要件によって異なります。

たとえば、サイトが突然停止した場合の基本的な高速DR手順は、次の基本的な手順を使用します。

- MetroClusterスイッチオーバーを強制的に実行する
- FC / iSCSI LUNの検出の実行（SANのみ）
- ファイルシステムをマウント
- SQLサービスの開始

このアプローチの主な要件は、リモートサイトでOSを実行することです。SQL Serverのセットアップ時に事前に設定し、同等のビルドバージョンで更新する必要があります。SQL Serverシステムデータベースは、災害が発生した場合にリモートサイトにミラーリングしてマウントすることもできます。

仮想データベースをホストするボリューム、ファイルシステム、およびデータストアが、スイッチオーバー前にディザスタリカバリサイトで使用されていない場合は、関連するボリュームで設定する必要はありません `dr-force-nvfail`。

SnapMirrorアクティブ同期

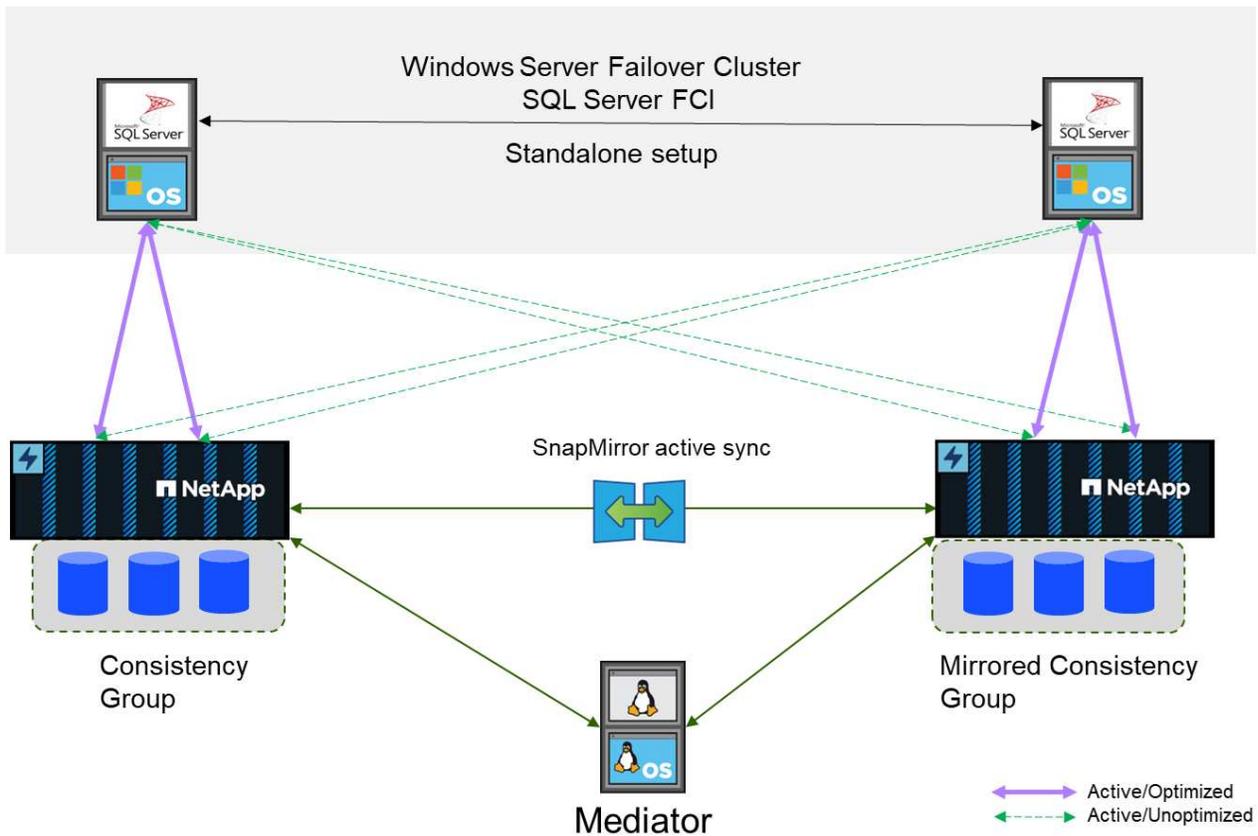
概要

SnapMirror Active Syncを使用すると、ストレージやネットワークが停止しても、個々のSQL Serverデータベースとアプリケーションの運用を継続できます。透過的なストレージフェイルオーバーにより、手動操作は不要です。

ONTAP 9.15.1以降、SnapMirrorアクティブ同期では、既存の非対称構成に加えて、対称アクティブ/アクティブアーキテクチャがサポートされます。対称アクティブ/アクティブ機能により、ビジネス継続性とディザスタリカバリのための同期双方向レプリケーションを提供します。複数の障害ドメインにわたるデータへの同時読み取り/書き込みアクセスにより、重要なSANワークロードのデータアクセスを保護し、運用を中断させず、災害やシステム障害時のダウンタイムを最小限に抑えることができます。

SQL Serverホストは、ファイバチャネル（FC）LUNまたはiSCSI LUNを使用してストレージにアクセスします。レプリケートされたデータのコピーをホストする各クラスタ間のレプリケーション。この機能はストレージレベルのレプリケーションであるため、スタンドアロンホストインスタンスまたはフェイルオーバークラスティンスタンス上で実行されているSQL Serverインスタンスは、どちらのクラスタでも読み取り/書き込み処理を実行できます。計画と設定の手順については、[を参照してください"SnapMirror Active Syncに関するONTAPドキュメント"](#)。

対称アクティブ/アクティブ構成の**SnapMirror**アクティブ同期アーキテクチャ



同期レプリケーション

通常運用時には、1つの例外を除き、各コピーは常にRPO=0の同期レプリカになります。データをレプリケートできない場合、ONTAPでは、データのレプリケートという要件が解除され、一方のサイトのLUNがオフラインになる間に、一方のサイトでIOの提供が再開されます。

ストレージハードウェア

他のストレージディザスタリカバリソリューションとは異なり、SnapMirrorアクティブ同期は非対称プラットフォームの柔軟性を提供します。各サイトのハードウェアが同一である必要はありません。この機能を使用すると、SnapMirrorアクティブ同期をサポートするために使用するハードウェアのサイズを適正化できます。リモートストレージシステムは、本番環境のワークロードを完全にサポートする必要がある場合はプライマリサイトと同一にすることができますが、災害によってI/Oが減少した場合は、リモートサイトの小規模システムよりも対費用効果が高くなります。

• ONTAPメディエーター**

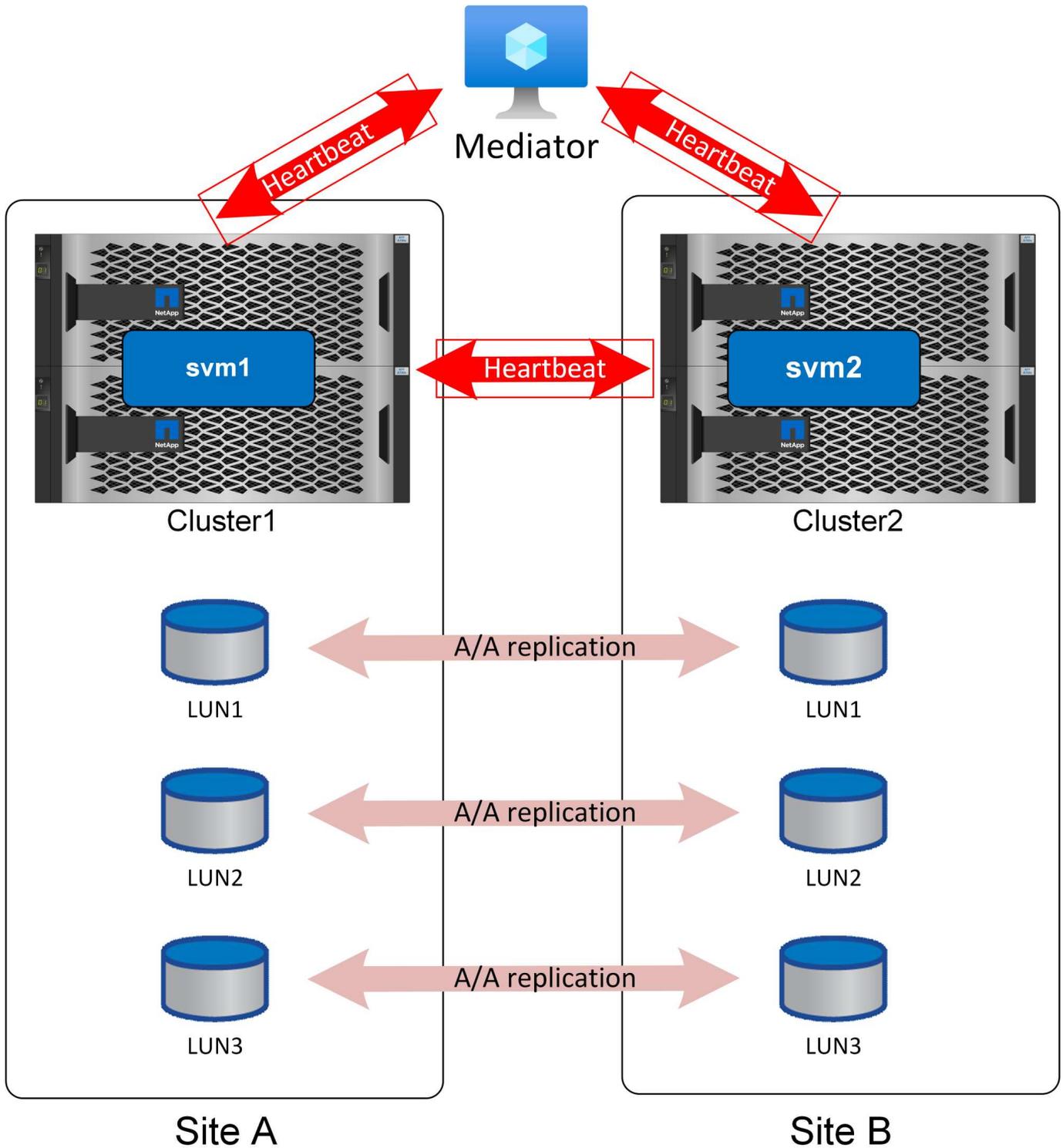
ONTAPメディエーターは、NetAppサポートからダウンロードするソフトウェアアプリケーションで、通常は小規模な仮想マシンに導入されます。ONTAPメディエーターはTiebreakerではありません。これは、SnapMirrorのアクティブな同期レプリケーションに含まれる2つのクラスタの代替通信チャンネルです。自動処理は、パートナーから直接接続またはメディエーター経由で受け取った応答に基づいてONTAPによって実行されます。

ONTAPメディエーター

フェイルオーバーを安全に自動化するにはMediatorが必要です。理想的には、独立した3つ目のサイトに配置しますが、レプリケーションに参加しているクラスタの1つと同じ場

所に配置すれば、ほとんどのニーズに対応できます。

調停者は実際にはタイブレーカーではありませんが、実質的にはそれがその機能を提供します。メディエーターは、クラスターノードの状態を判断するのに役立ち、サイト障害が発生した場合の自動切り替えプロセスを支援します。Mediator はいかなる状況でもデータを転送しません。



自動フェイルオーバーの最大の課題はスプリットブレインの問題であり、この問題は2つのサイト間の接続が失われた場合に発生します。何が起るべきでしょうか？2つの異なるサイトがデータのサバイバーコピーとして自分自身を指定する必要はありませんが、1つのサイトでは、反対側のサイトが実際に失われたことと、

反対側のサイトと通信できないことを区別するにはどうすればよいでしょうか。

ここでメディアエーターが写真に入ります3番目のサイトに配置され、各サイトからそのサイトへの個別のネットワーク接続がある場合は、他のサイトの正常性を検証するための追加のパスが各サイトに用意されています。上の図をもう一度見て、次のシナリオを検討してください。

- 一方または両方のサイトからメディアエーターに障害が発生した場合、またはメディアエーターに到達できない場合はどうなりますか？
 - 2つのクラスタは、レプリケーションサービスに使用されるのと同じリンクを介して相互に通信できません。
 - データは引き続きRPO=0の保護で提供される
- サイトAに障害が発生した場合の動作
 - サイトBは、両方の通信チャンネルがダウンしたことを確認します。
 - サイトBがデータサービスをテイクオーバーするが、RPO=0ミラーリングなし
- サイトBで障害が発生した場合の動作
 - サイトAでは、両方の通信チャンネルがダウンしていることが確認されます。
 - サイトAがデータサービスをテイクオーバーするが、RPO=0ミラーリングなし

もう1つ考慮すべきシナリオがあります。データレプリケーションリンクの停止です。サイト間のレプリケーションリンクが失われた場合、RPO=0のミラーリングは明らかに不可能です。ではどうすればいいのでしょうか。

これは、優先サイトのステータスによって制御されます。SM-AS関係では、一方のサイトがもう一方のサイトのセカンダリになります。これは通常の運用には影響せず、すべてのデータアクセスは対称的ですが、レプリケーションが中断された場合は、運用を再開するためにこの関係を解除する必要があります。その結果、優先サイトはミラーリングなしで処理を継続し、レプリケーション通信がリストアされるまでセカンダリサイトはIO処理を停止します。

優先サイト

SnapMirrorのアクティブな同期の動作は対称ですが、重要な例外が1つあります（推奨サイト構成）。

SnapMirrorアクティブ同期では、一方のサイトが「ソース」で、もう一方が「デスティネーション」と見なされます。これは一方向のレプリケーション関係を意味しますが、IO動作には適用されません。レプリケーションは双方向であり、対称であり、IO応答時間はミラーの両側で同じです。

「source」指定は、優先サイトを制御します。レプリケーションリンクが失われた場合、ソースコピー上のLUNパスは引き続きデータを提供しますが、デスティネーションコピー上のLUNパスは、レプリケーションが再確立されてSnapMirrorが同期状態に戻るまで使用できなくなります。その後、パスでデータの提供が再開されます。

ソース/デスティネーションの設定はSystemManagerで確認できます。

Relationships

Local destinations Local sources

Search Download Show/hide: Filter

Source	Destination	Policy type
▼ jfs_as1:/cg/jfsAA	jfs_as2:/cg/jfsAA	Synchronous

または、CLIで次の操作を行います。

```
Cluster2::> snapmirror show -destination-path jfs_as2:/cg/jfsAA

          Source Path: jfs_as1:/cg/jfsAA
          Destination Path: jfs_as2:/cg/jfsAA
          Relationship Type: XDP
          Relationship Group Type: consistencygroup
          SnapMirror Schedule: -
          SnapMirror Policy Type: automated-failover-duplex
          SnapMirror Policy: AutomatedFailOverDuplex
          Tries Limit: -
          Throttle (KB/sec): -
          Mirror State: Snapmirrored
          Relationship Status: InSync
```

重要なのは、ソースがcluster1のSVMであることです。前述のように、「ソース」と「デスティネーション」という用語は、レプリケートされたデータのフローを表していません。両方のサイトが書き込みを処理し、反対側のサイトにレプリケートできます。実際には、両方のクラスタがソースとデスティネーションです。1つのクラスタをソースとして指定すると、レプリケーションリンクが失われた場合に、どのクラスタが読み取り/書き込みストレージシステムとして残っているかが制御されます。

ネットワークポロジ

均一なアクセス

統一されたアクセスネットワークとは、ホストが両方のサイト（または同じサイト内の障害ドメイン）のパスにアクセスできることを意味します。

SM-ASの重要な機能の1つは、ホストがどこにあるかを認識するようにストレージシステムを設定できることです。LUNを特定のホストにマッピングするときに、LUNが特定のストレージシステムに近接しているかどうかを指定できます。

近接設定

プロキシミティとは、特定のホストWWNまたはiSCSIイニシエータIDがローカルホストに属していることを

示すクラスタ単位の構成を指します。これは、LUNアクセスを設定するための2番目のオプションの手順です。

最初の手順では、通常のigroup設定を行います。各LUNは、そのLUNにアクセスする必要があるホストのWWN/iSCSI IDを含むigroupにマッピングする必要があります。これは、どのホストがLUNに_access_toを持つかを制御します。

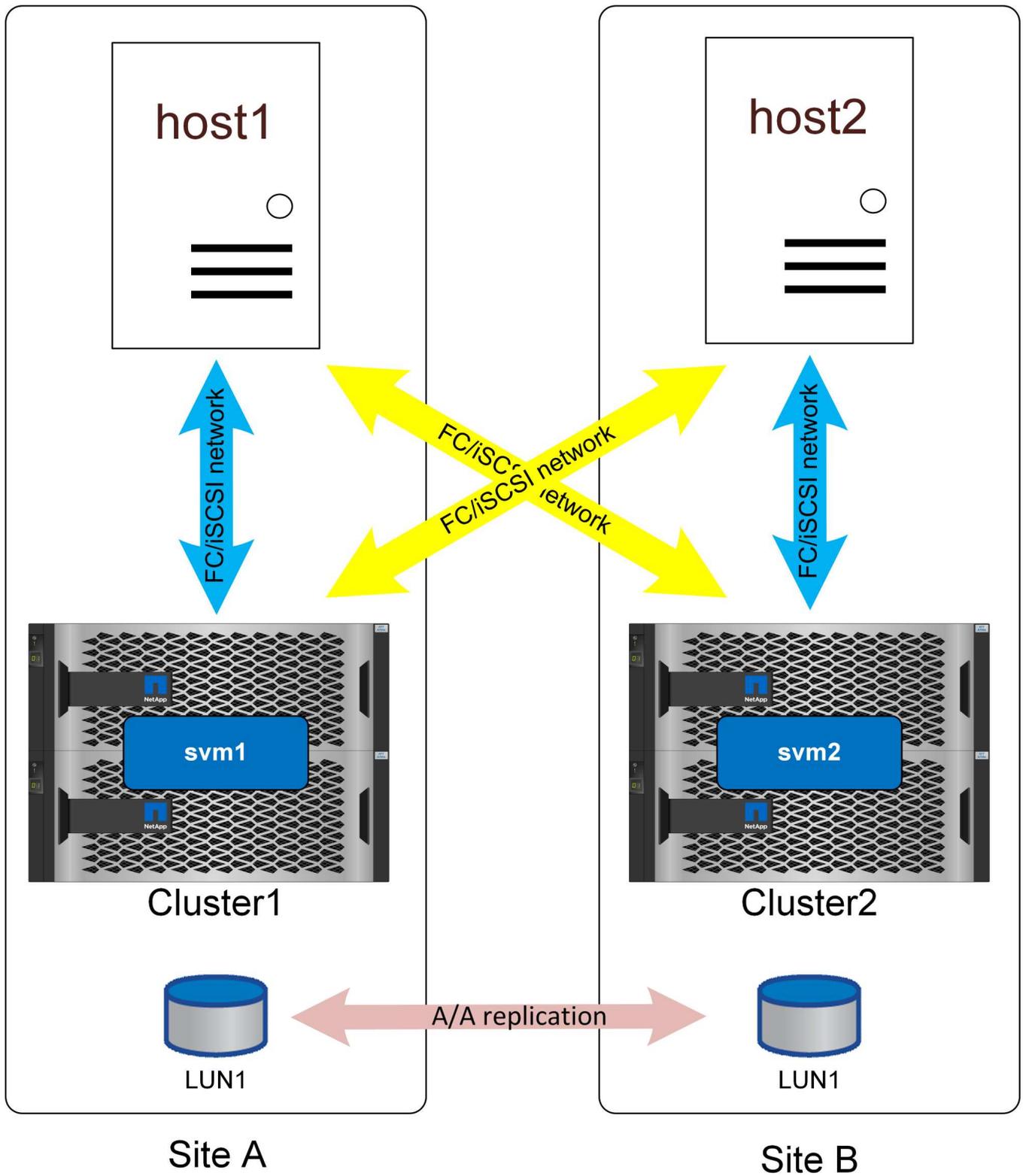
2番目のオプション手順は、ホストプロキシミティを設定することです。これはアクセスを制御するのではなく、_priority_を制御します。

たとえば、サイトAのホストがSnapMirror Active Syncで保護されているLUNにアクセスするように設定されている場合、SANがサイト間で拡張されるため、サイトAのストレージまたはサイトBのストレージを使用してそのLUNへのパスを使用できます。

近接設定を使用しない場合、両方のストレージシステムがアクティブな最適パスをアダプタイズするため、そのホストは両方のストレージシステムを均等に使用します。SANのレイテンシやサイト間の帯域幅に制限がある場合は、この設定を解除できない可能性があります。また、通常動作中に各ホストがローカルストレージシステムへのパスを優先的に使用するよう設定することもできます。これは、ホストWWN/iSCSI IDをローカルクラスタに近接ホストとして追加することで設定します。これは、CLIまたはSystemManagerで実行できます。

AFF

AFFシステムでホストプロキシミティが設定されている場合、パスは次のように表示されます。



Active/Optimized Path

Active Path

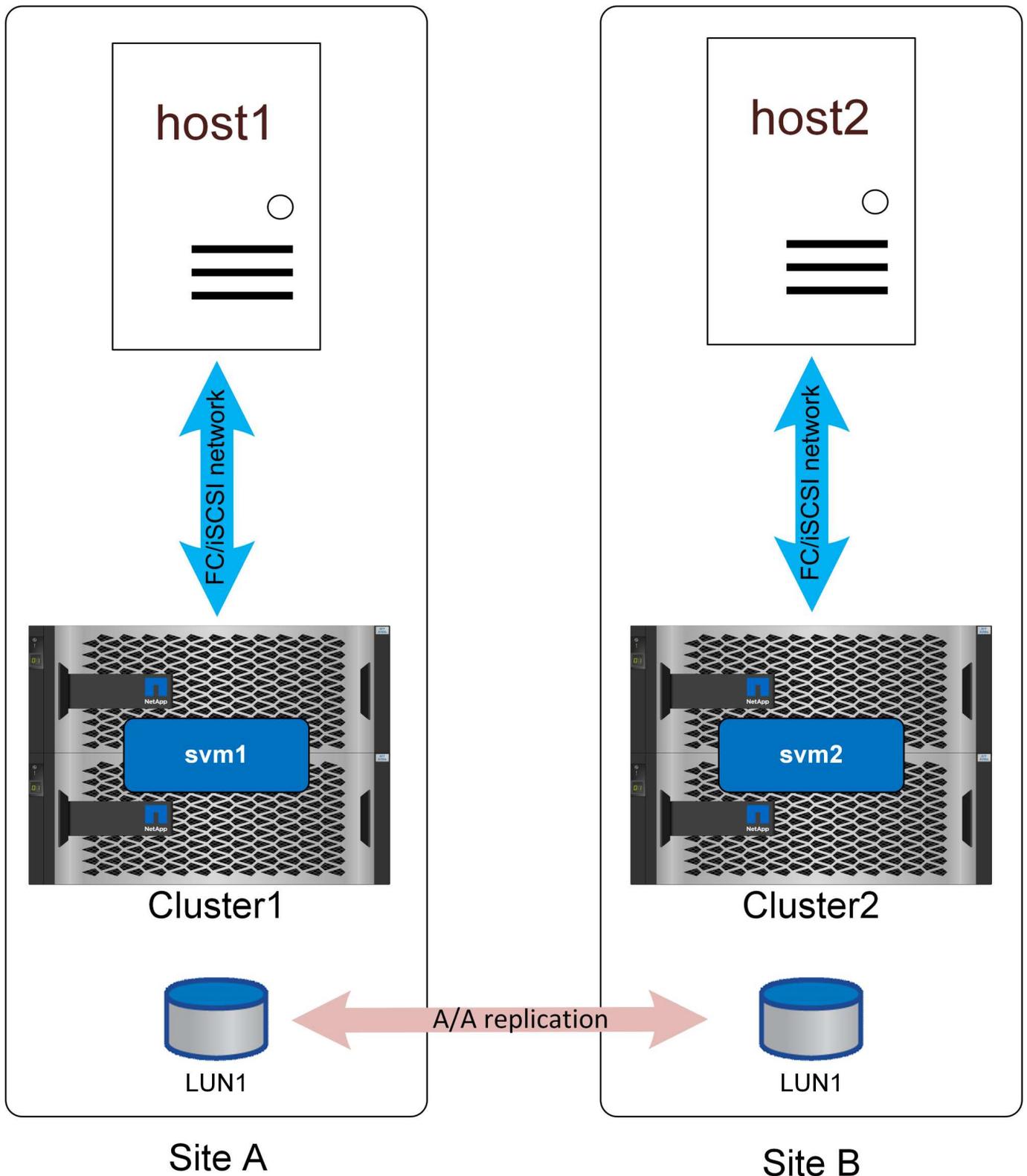
通常の運用では、すべてのIOがローカルIOになります。読み取りと書き込みはローカルストレージアレイから処理されます。もちろん、書き込みIOも確認応答の前にローカルコントローラでリモートシステムにレプリケートする必要がありますが、すべての読み取りIOはローカルで処理されるため、サイト間のSANリンクを経由して余分なレイテンシが発生することはありません。

非最適パスが使用されるのは、すべてのアクティブ/最適パスが失われた場合だけです。たとえば、サイトAのアレイ全体に電力が供給されなくなっても、サイトAのホストはサイトBのアレイへのパスに引き続きアクセスできるため、レイテンシは高くなりますが運用を継続できます。

この図では、わかりやすいように、ローカルクラスタを経由する冗長パスを示していません。ONTAPストレージシステム自体はHAであるため、コントローラ障害が発生してもサイト障害は発生しません。影響を受けるサイトで使用されるローカルパスが変更されるだけです。

不均一なアクセス

非ユニフォームアクセスネットワークとは、各ホストがローカルストレージシステム上のポートにしかアクセスできないことを意味します。SANを複数のサイト（または同じサイト内の障害ドメイン）に拡張することはできません。



Active/Optimized Path

このアプローチの主なメリットはSANのシンプルさです。SANをネットワーク経由で拡張する必要がなくなります。お客様によっては、サイト間の接続遅延が十分でない場合や、サイト間ネットワーク経由でFC SAN

トラフィックをトンネリングするためのインフラストラクチャが不足している場合があります。

不均一なアクセスの欠点は、レプリケーションリンクの喪失などの特定の障害シナリオで、一部のホストがストレージにアクセスできなくなることです。ローカルストレージの接続が失われると、単一のホストでのみ実行されている非クラスタデータベースなど、単一インスタンスとして実行されるアプリケーションは失敗します。データは保護されますがデータベース・サーバはアクセスできなくなりますリモートサイトで、できれば自動化されたプロセスを使用して再起動する必要があります。たとえば、VMware HAは、あるサーバでオールパスダウン状態を検出し、パスが使用可能な別のサーバでVMを再起動できます。

一方、Oracle RACなどのクラスタ化されたアプリケーションは、2つの異なるサイトで同時に利用可能なサービスを提供できます。サイトが失われても、アプリケーションサービス全体が失われるわけではありません。サバイバースイトでは、引き続きインスタンスを使用して実行できます。

多くの場合、サイト間リンク経由でストレージにアクセスするアプリケーションによるレイテンシのオーバーヘッドは許容できません。つまり、サイトのストレージが失われると、障害が発生したサイトのサービスをシャットダウンする必要が生じるため、統一されたネットワークの可用性の向上は最小限で済みます。



この図では、わかりやすいように、ローカルクラスタを経由する冗長パスを示していません。ONTAPストレージシステム自体はHAであるため、コントローラ障害が発生してもサイト障害は発生しません。影響を受けるサイトで使用されるローカルパスが変更されるだけです。

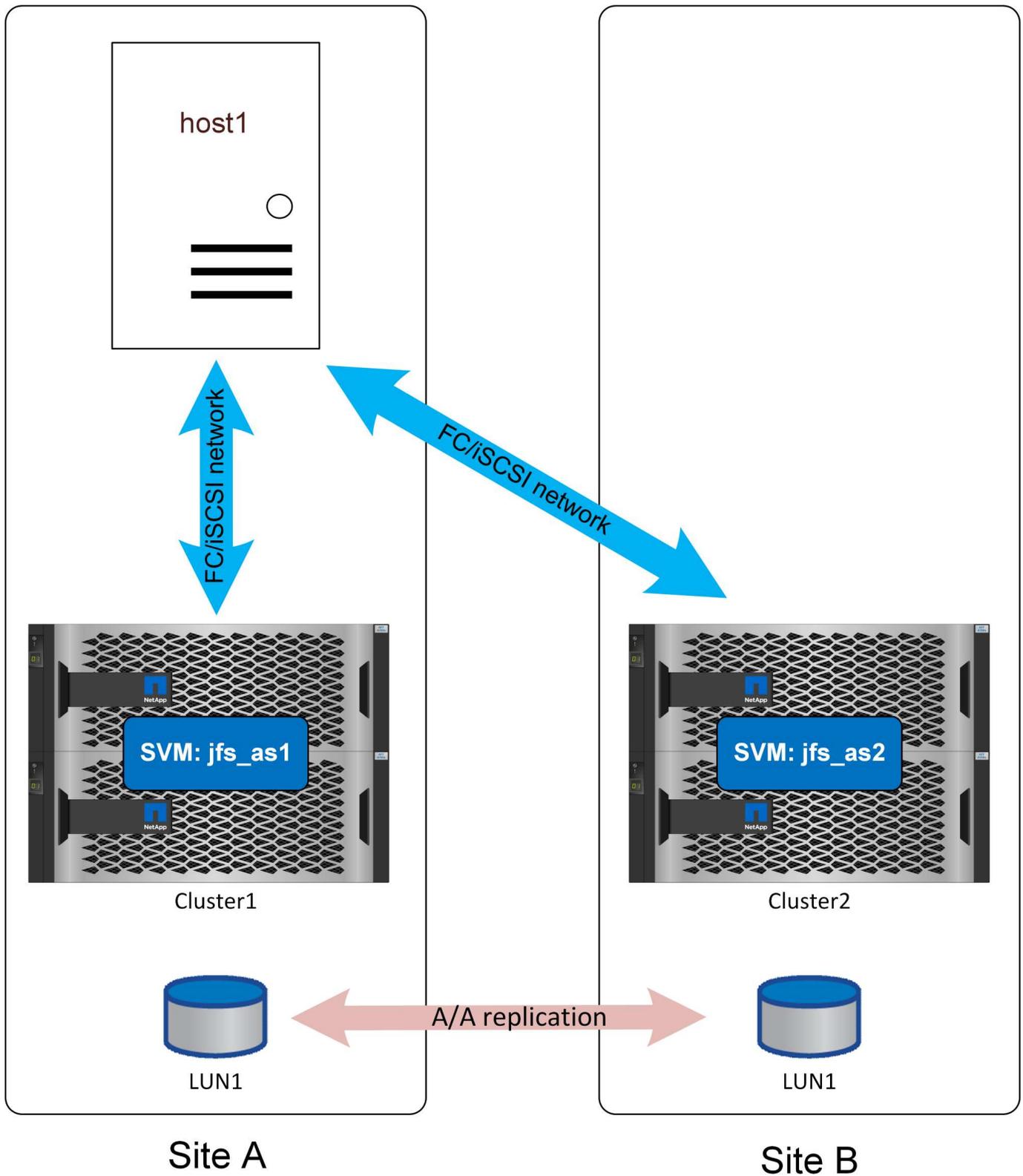
概要

SQL Serverは、いくつかの方法でSnapMirrorアクティブ同期と連携するように設定できます。適切な答えは、使用可能なネットワーク接続、RPOの要件、可用性の要件によって異なります。

SQL Serverのスタンドアロンインスタンス

ファイルレイアウトとサーバ設定のベストプラクティスは、ドキュメントで推奨されているものと同じ"[ONTAP上のSQL Server](#)"です。

スタンドアロン構成では、SQL Serverを1つのサイトでのみ実行できます。おそらく"[均一 \(Uniform\)](#)"アクセスが使用されます。



アクセス方法が統一されていれば、どちらかのサイトでストレージ障害が発生してもデータベースの処理は中断されません。データベース・サーバを含むサイトで完全なサイト障害が発生すると'当然'システム停止が発生します

一部のお客様は、リモートサイトで実行されているOSを、構成済みのSQL Serverセットアップで構成し、本番インスタンスと同等のビルドバージョンで更新することもできます。フェイルオーバーを実行するには、代

替サイトでSQL Serverのスタンドアロンインスタンスをアクティブ化し、LUNを検出して、データベースを起動する必要があります。ストレージ側からの操作が不要なため、Windows PowerShellコマンドレットを使用して完全なプロセスを自動化できます。

"不均一"アクセスも使用できますが、データベースにストレージへの使用可能なパスがないために、データベースサーバが配置されていたストレージシステムで障害が発生すると、データベースが停止します。これは、場合によっては許容される可能性があります。SnapMirrorのアクティブな同期では引き続きRPO=0のデータ保護が提供され、サイト障害が発生した場合でも、稼働しているコピーがアクティブになり、前述の統一されたアクセスと同じ手順で運用を再開できます。

シンプルで自動化されたフェイルオーバープロセスは、仮想化ホストを使用してより簡単に設定できます。たとえば、SQL ServerデータファイルをブートVMDKとともにセカンダリストレージに同期的にレプリケートする場合は、災害が発生したときに代替サイトで環境全体をアクティブ化できます。管理者は、サバイバーサイトでホストを手動でアクティブ化することも、VMware HAなどのサービスを使用してプロセスを自動化することもできます。

SQL Serverフェイルオーバークラスターインスタンス

SQL Serverフェイルオーバーインスタンスは、物理サーバまたは仮想サーバ上でゲストオペレーティングシステムとして実行されているWindowsフェイルオーバークラスターでホストすることもできます。このマルチホストアーキテクチャは、SQL Serverインスタンスとストレージの耐障害性を提供します。このような導入は、強化されたパフォーマンスを維持しながら堅牢なフェイルオーバープロセスを必要とする、負荷の高い環境に役立ちます。フェイルオーバークラスターのセットアップでは、ホストまたはプライマリストレージが影響を受けると、SQLサービスがセカンダリホストにフェイルオーバーされ、同時にセカンダリストレージがIOを処理できるようになります。自動化スクリプトや管理者の介入は必要ありません。

障害シナリオ

完全なSnapMirrorアクティブ同期アプリケーションアーキテクチャを計画するには、さまざまな計画的フェイルオーバーシナリオと計画外フェイルオーバーシナリオでSM-ASがどのように対応するかを理解する必要があります。

次の例では、サイトAが優先サイトとして設定されているとします。

レプリケーション接続の切断

SM-ASレプリケーションが中断されると、クラスターが反対側のサイトに変更をレプリケートできなくなるため、書き込みIOを完了できません。

サイトA（優先サイト）

優先サイトでのレプリケーションリンク障害の結果、レプリケーションリンクが本当に到達不能であると判断される前に、ONTAPがレプリケートされた書き込み処理を再試行するため、書き込みIO処理が約15秒間中断されます。15秒が経過すると、サイトAのシステムが読み取りと書き込みのIO処理を再開します。SANパスは変更されず、LUNはオンラインのままです。

サイトB

サイトBはSnapMirrorアクティブ同期優先サイトではないため、約15秒後にLUNパスが使用できなくなります。

ストレージシステムの障害

ストレージシステム障害の結果は、レプリケーションリンクが失われた場合とほぼ同じです。サバイバーサイトでは、IOが約15秒間停止します。その15秒が経過すると、IOは通常どおりそのサイトで再開されます。

メディエーターの停止

メディエーターサービスはストレージの処理を直接制御しません。クラスタ間の代替制御パスとして機能します。これは主に、スプリットブレインのリスクを伴わずにフェイルオーバーを自動化することを目的としています。通常運用時は、各クラスタがパートナーに変更内容をレプリケートするため、各クラスタはパートナークラスタがオンラインでデータを提供していることを確認できます。レプリケーションリンクに障害が発生すると、レプリケーションは停止します。

安全な自動フェイルオーバーを実現するためにメディエーターが必要になるのは、そうしないと、双方向通信の切断がネットワークの停止によるものか実際のストレージ障害によるものかをストレージクラスタが判断できないためです。

メディエーターは、パートナーの健全性を確認するための代替パスを各クラスタに提供します。シナリオは次のとおりです。

- クラスタがパートナーに直接接続できる場合は、レプリケーションサービスが動作しています。対処は不要です。
- 優先サイトがパートナーに直接またはメディエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離されてLUNパスがオフラインになっているとみなされます。その後、優先サイトでRPO=0の状態が解除され、読み取りI/Oと書き込みI/Oの両方の処理が継続されます。
- 非優先サイトがパートナーに直接接続できず、メディエーター経由で接続できる場合、そのサイトのパスはオフラインになり、レプリケーション接続が戻るまで待機します。
- 優先されないサイトがパートナーに直接、または動作中のメディエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離され、LUNパスがオフラインになったとみなされます。優先されないサイトは、RPO=0状態の解放に進み、読み取りI/Oと書き込みI/Oの両方の処理を続行します。レプリケーションソースの役割を引き継ぎ、新しい優先サイトになります。

メディエーターが完全に使用できない場合：

- 非優先サイトまたはストレージシステムの障害など、何らかの理由でレプリケーションサービスに障害が発生すると、優先サイトでRPO=0状態が解放され、読み取りおよび書き込みIO処理が再開されます。非優先サイトのパスがオフラインになります。
- 優先サイトに障害が発生すると、非優先サイトでは、反対側のサイトが本当にオフラインであることを確認できず、そのため非優先サイトがサービスを再開しても安全ではないため、システムが停止します。

サービスのリストア

サイト間の接続のリストアや障害が発生したシステムの電源投入などの障害が解決されると、SnapMirrorのアクティブな同期エンドポイントは、障害のあるレプリケーション関係の存在を自動的に検出してRPO=0状態に戻します。同期レプリケーションが再確立されると、障害が発生したパスは再びオンラインになります。

多くの場合、クラスタ化されたアプリケーションは障害が発生したパスの復帰を自動的に検出し、それらのアプリケーションもオンラインに戻ります。また、ホストレベルのSANスキャンが必要な場合や、アプリケーションを手動でオンラインに戻す必要がある場合もあります。それはアプリケーションとそれがどのように構成されているかによって異なり、一般的にそのようなタスクは簡単に自動化することができます。ONTAP自体は自己回復型であり、RPO=0のストレージ処理を再開するためにユーザの介入は不要です。

手動フェイルオーバー

優先サイトを変更するには、簡単な操作が必要です。クラスタ間でレプリケーション動作の権限が切り替わるため、IOは1~2秒間停止しますが、それ以外の場合はIOには影響しません。

ASA R2システムでのストレージ構成

概要

NetApp ASA R2は、ミッションクリティカルなワークロードを実行するSANのみのお客様向けの、シンプルで強力なソリューションです。ONTAPストレージソリューションを実行するASA R2プラットフォームと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が可能になります。

次のASAプラットフォームは、すべてのSANプロトコル (iSCSI、FC、NVMe/FC、NVMe/TCP) をサポートするASA R2システムに分類されます。iSCSI、FC、NVMe/FC、NVMe/TCPの各プロトコルでは、対称アクティブ/アクティブアーキテクチャのマルチパスがサポートされているため、ホストとストレージの間のすべてのパスがアクティブ/最適化されます。

- ASAA1K
- ASAA90
- ASAA70
- ASAA50
- ASAA30
- ASAA20

詳細については、[を参照してください。 "NetApp ASA"](#)

SQL Server on ONTAPソリューションを最適化するには、SQL ServerのI/Oパターンと特性を理解する必要があります。SQL Serverデータベース用のストレージレイアウトを適切に設計するには、SQL Serverのパフォーマンス要件を満たしながら、インフラ全体の管理性を最大限に高める必要があります。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

Microsoftでは、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ています。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"](#)。

ASAのストレージユニットとは、SCSI/FCホストの場合はLUN、NVMeホストの場合はNVMeネームスペースを指します。サポートされるプロトコルに基づいて、LUN、NVMeネームスペース、またはその両方を作成するように求められます。データベース導入用のストレージユニットを作成する前に、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。ストレージユニットに関する次のNetAppの推奨事項を参照してください。

- 管理の複雑さを避けるため、同じホスト上で実行される複数のSQL Server間で同じストレージユニットを共有しないでください。同じホストで複数のSQL Serverインスタンスを実行する場合は、ノードのストレージユニット数の上限に近い場合を除き、共有は避け、データ管理を容易にするために、ホストごとにインスタンスごとにストレージユニットを個別に配置してください。
- 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。
- 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 データストレージユニットのSnapshotコピーを調整します。
- SQL Serverシステムデータベースは専用のストレージユニットに配置します。
- 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専用のストレージユニットに配置します。ストレージユニットの数が課題となる大規模な環境では、慎重に計画したあとに、tempdbとシステムデータベースを同じストレージユニットに統合できます。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。
- (.mdf、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を別のストレージユニットに配置します。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ストレージユニットまたはVMDKに配置し、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。また、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とデータファイルのランダムリード/ライトI/Oが分離され、SQL Serverのパフォーマンスが大幅に向上します。
- 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は、バックアップポリシーのSnapMirror機能でSnapshotが保持ポリシーによって上書きされないように、別々のストレージユニットに配置してください。
- データベースファイルとデータベース以外のファイル（フルテキスト検索関連ファイルなど）を同じストレージユニットに混在させないでください。
- 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）別のストレージユニットに配置すると、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、データベースのファイルがそのストレージユニットを他のファイルと共有していない`.mdf`場合にのみ有効`.mdf`です。
- Windowsサーバのディスクマネージャを使用してディスクをフォーマットするときは、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。
- マウントポイントをホストするストレージユニット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。
- 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とネイティブ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。
- Always Onフェイルオーバークラスターインスタンスを使用している場合は、Windowsサーバフェイルオーバークラスターノード間で共有されるストレージユニットにユーザデータベースを配置する必要があります。また、物理ディスククラスターリソースは、SQL Serverインスタンスに関連付けられたクラスターグループに割り当てられます。

データベースファイルとファイルグループ

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。

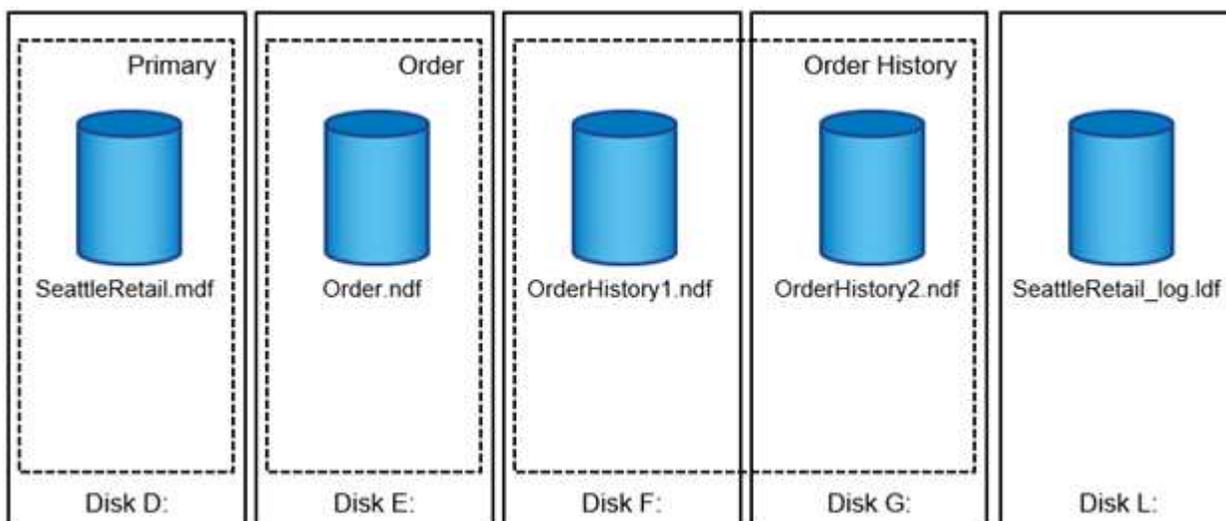
理論的には、SQL Server (64ビット) ではインスタンスあたり32、767個のデータベースと524、272TBのデータベースサイズがサポートされますが、通常のインストールでは複数のデータベースが使用されます。ただし、SQL Serverで処理できるデータベースの数は、負荷とハードウェアによって異なります。SQL Serverインスタンスでは、数十、数百、場合によっては数千の小規模データベースをホストしていることも珍しくありません。

データベースファイルとファイルグループ

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。与えられたワークロードをサポートするデータファイルの数は、アプリケーションに影響を与えることなく、I/O要件と想定容量をサポートするために必要に応じて変えることができます。データベースレイアウトのバリエーションは、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。

データベースインスタンスファイルの初期化

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に、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。



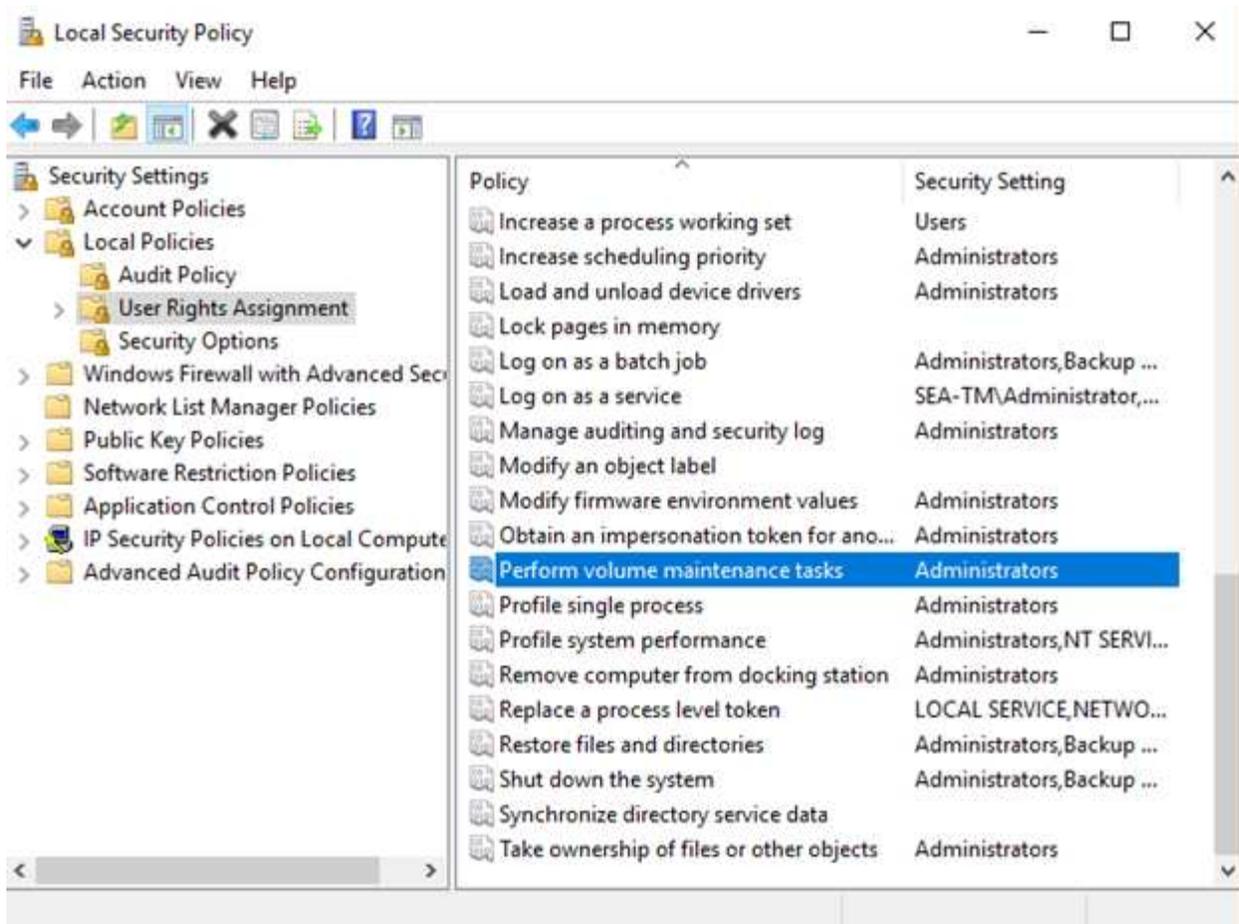
* NetAppでは、1つのファイルグループ内のすべてのファイルに同じ初期サイズと自動拡張パラメータを設定し、拡張サイズをパーセンテージではなくメガバイト単位で定義することを推奨しています*。これにより、Proportional Fill Algorithmは、データファイル間で書き込みアクティビティのバランスを均等に調整できます。

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。



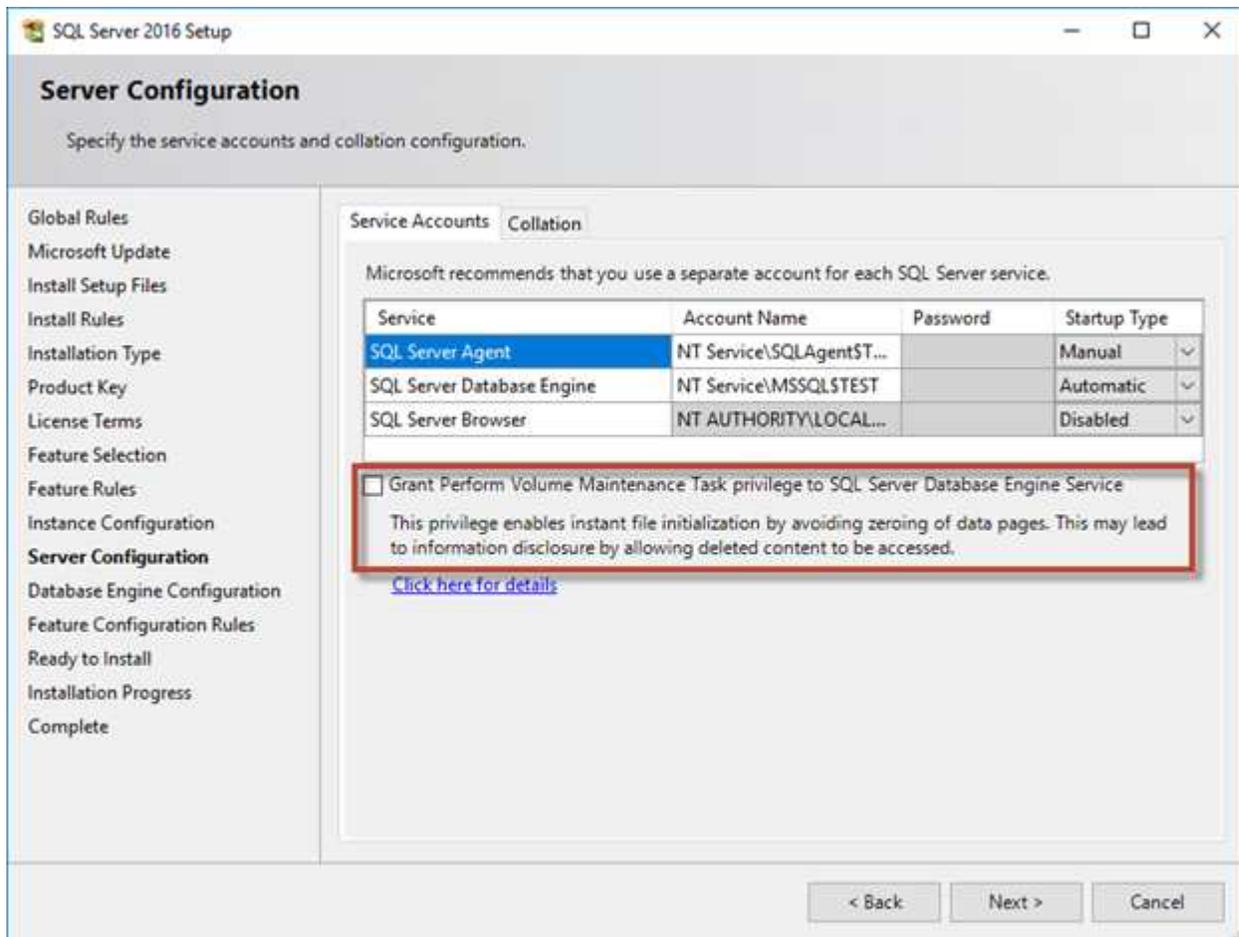
権限が有効になっているかどうかを確認するには、次の例のコードを使用します。このコードは、SQL Serverがエラーログに追加情報を書き込み、小さなデータベースを作成し、ログの内容を読み取るように強制する2つのトレースフラグを設定します。

```
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.

[Perform Volume Maintenance]タスクはSQL Server 2016では簡素化され、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。

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 사용되는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。

ログディレクトリを専用のストレージユニットに配置します。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。
- 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ストレージ・ユニットに、ホスト・ログ・ディレクトリを作成します。
- Always On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、ホストログディレクトリに使用されるストレージユニットは、SnapCenterでバックアップされるSQL Serverインスタンスと同じクラスターグループ内のクラスターディスクリソースである必要があります。

データ保護

データベースのバックアップ戦略は、理論的な機能ではなく、特定されたビジネス要件に基づいて行う必要があります。ONTAPのSnapshotテクノロジーとMicrosoft SQL Server APIを組み合わせることで、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。

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、ストレージユニット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. データベースワークロードが複数のストレージユニットにまたがっている場合は、整合グループを作成して管理タスクを簡易化します。整合グループは、単一ユニットとして管理されるストレージユニットの集まりです。
4. 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。
5. 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](#)"。



* 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。前述の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。

ディザスタリカバリ

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には、データ保護とAlways-Onを統合するためのオプションが用意されています。ただし、場合によっては、次のオプションを使用してONTAPレプリケーションテクノロジーを検討することもできます。

SnapMirror

SnapMirrorテクノロジーは、LANおよびWAN経由でデータを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。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。同期モードまたは非同期モードのいずれかで設定できます。NetApp ASAでのSnapMirror同期レプリケーションは、SnapMirrorアクティブ同期を使用して設定します。

SnapMirrorアクティブ同期

多くのお客様にとって、ビジネス継続性には、単にデータのリモートコピーを所有するだけでなく、SnapMirrorのアクティブ同期を使用してNetApp ONTAPで可能なデータを迅速に利用できる機能が必要です。

SnapMirrorアクティブ同期を使用すると、基本的には2つの異なるONTAPシステムでLUNデータの独立したコピーを維持しながら、このLUNの単一インスタンスを提供できます。ホストの観点からは、単一のLUNエンティティです。SnapMirrorアクティブ同期は、iSCSI / FCベースのLUNでサポートされます。

SnapMirrorアクティブ同期はRPO=0のレプリケーションを提供し、2つの独立したクラスタ間で簡単に実装できます。データの2つのコピーが同期されると、2つのクラスタは書き込みをミラーリングするだけで済みます。一方のクラスタで書き込みが発生すると、もう一方のクラスタにレプリケートされます。書き込みの確認応答がホストに送信されるのは、両方のサイトで書き込みが完了した場合だけです。このプロトコルスプリット動作以外では、2つのクラスタは通常のONTAPクラスタです。

SM-ASの主なユースケースの1つに、きめ細かなレプリケーションがあります。すべてのデータを1つのユニットとしてレプリケートしたくない場合や、特定のワークロードを選択的にフェイルオーバーできる必要があります。

SM-ASのもう1つの主なユースケースは、アクティブ/アクティブ処理です。アクティブ/アクティブ処理では、データの完全に使用可能なコピーを、同じパフォーマンス特性を持つ2つの異なるクラスタに配置し、必要に応じてSANをサイト間で拡張する必要がありません。アプリケーションがサポートされていれば、両方のサイトでアプリケーションをすでに実行しておくことができます。これにより、フェイルオーバー処理中の全体的なRTOが削減されます。

SnapMirror

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

- 迅速なデータリカバリのニーズが高い場合は、SnapMirrorアクティブ同期を使用した同期レプリケーションを使用し、RPOの柔軟性を高める非同期ソリューションを使用します。
- SnapCenterを使用してデータベースをバックアップし、Snapshotをリモートクラスタにレプリケートする場合は、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データを含むストレージユニット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』](#)"。

SnapMirrorアクティブ同期

概要

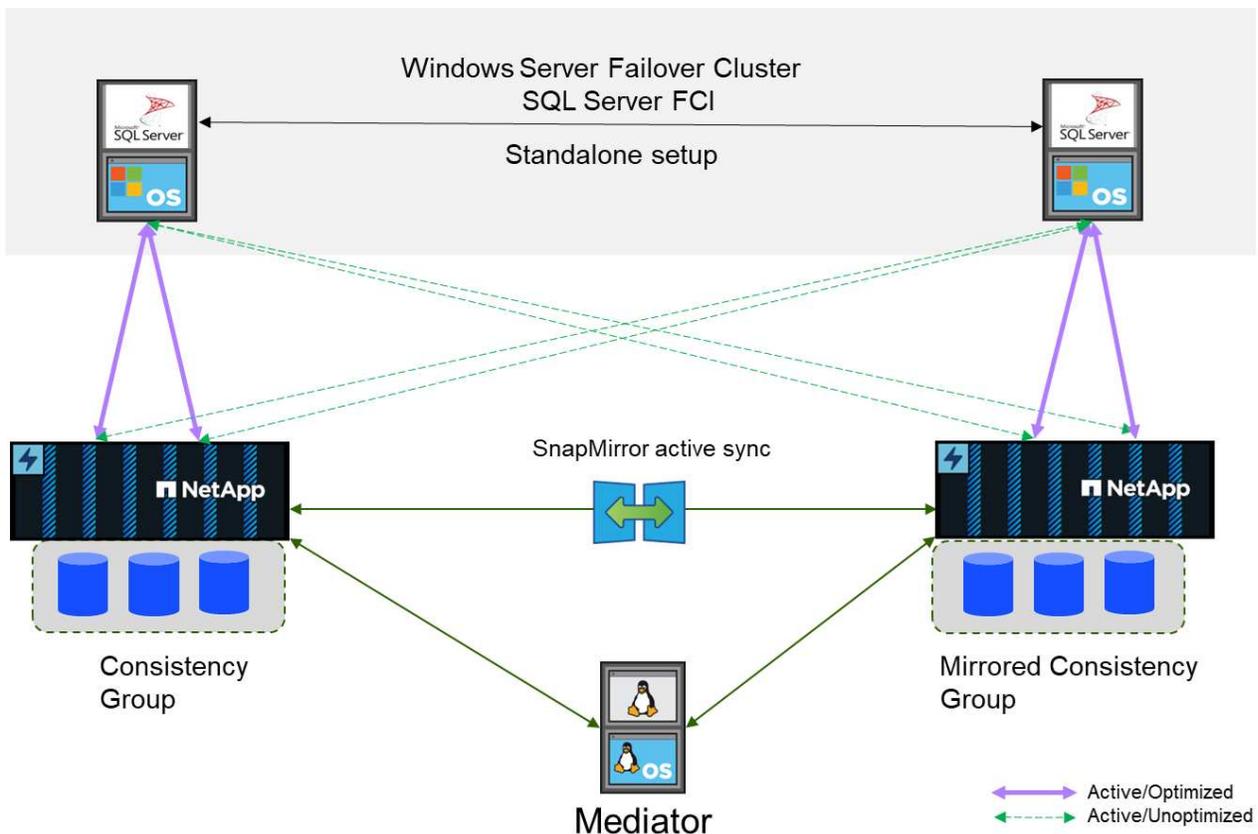
SnapMirror Active Syncを使用すると、ストレージやネットワークが停止しても、個々のSQL Serverデータベースとアプリケーションの運用を継続できます。透過的なストレージフェイルオーバーにより、手動操作は不要です。

SnapMirrorアクティブ同期は、同期双方向レプリケーションを提供する対称アクティブ/アクティブアーキテ

クチャをサポートし、ビジネス継続性とディザスタリカバリを実現します。複数の障害ドメインにわたるデータへの同時読み取り/書き込みアクセスにより、重要なSANワークロードのデータアクセスを保護し、運用を中断させず、災害やシステム障害時のダウンタイムを最小限に抑えることができます。

SQL Serverホストは、ファイバチャネル (FC) LUNまたはiSCSI LUNを使用してストレージにアクセスします。レプリケートされたデータのコピーをホストする各クラスタ間のレプリケーション。この機能はストレージレベルのレプリケーションであるため、スタンドアロンホストインスタンスまたはフェイルオーバークラスターインスタンス上で実行されているSQL Serverインスタンスは、どちらのクラスタでも読み取り/書き込み処理を実行できます。計画と設定の手順については、を参照してください"[SnapMirror Active Syncに関するONTAPドキュメント](#)"。

対称アクティブ/アクティブとSnapMirrorアクティブのアーキテクチャ



同期レプリケーション

通常運用時には、1つの例外を除き、各コピーは常にRPO=0の同期レプリカになります。データをレプリケートできない場合、ONTAPでは、データのレプリケートという要件が解除され、一方のサイトのLUNがオフラインになる間に、一方のサイトでIOの提供が再開されます。

ストレージハードウェア

他のストレージディザスタリカバリソリューションとは異なり、SnapMirrorアクティブ同期は非対称プラットフォームの柔軟性を提供します。各サイトのハードウェアが同一である必要はありません。この機能を使用すると、SnapMirrorアクティブ同期をサポートするために使用するハードウェアのサイズを適正化できます。リモートストレージシステムは、本番環境のワークロードを完全にサポートする必要がある場合はプライマリサイトと同一にすることができますが、災害によってI/Oが減少した場合は、リモートサイトの小規模システムよりも対費用効果が高くなります。

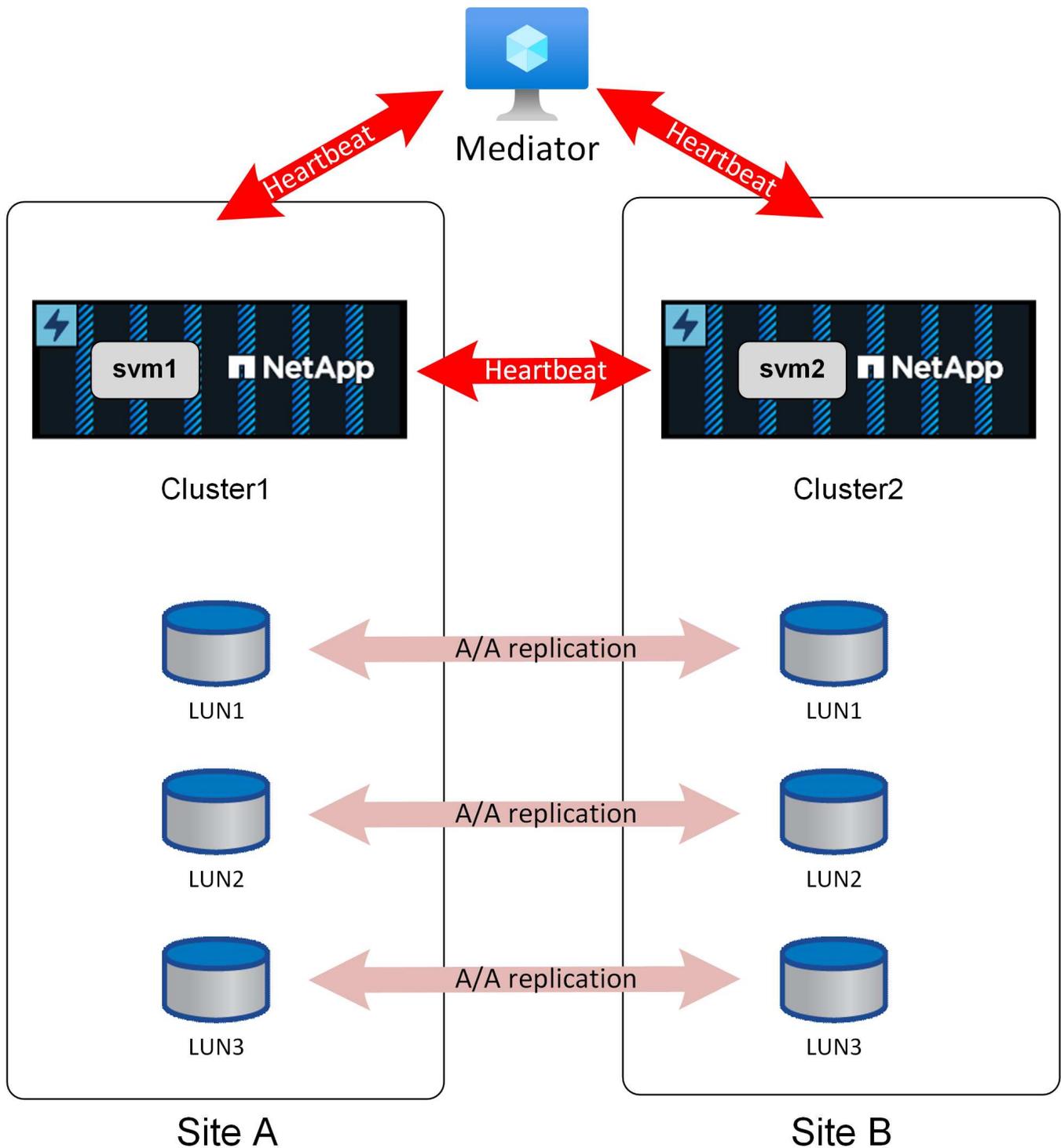
- ONTAPメディエーター**

ONTAPメディエーターは、NetAppサポートからダウンロードするソフトウェアアプリケーションで、通常は小規模な仮想マシンに導入されます。ONTAPメディエーターはTiebreakerではありません。これは、SnapMirrorのアクティブな同期レプリケーションに含まれる2つのクラスタの代替通信チャンネルです。自動処理は、パートナーから直接接続またはメディエーター経由で受け取った応答に基づいてONTAPによって実行されます。

ONTAPメディエーター

フェイルオーバーを安全に自動化するにはメディエーターが必要です。理想的には、独立した3つ目のサイトに配置しますが、レプリケーションに参加しているクラスタの1つと同じ場所に配置すれば、ほとんどのニーズに対応できます。

メディエーターは実際にはTiebreakerではありませんが、それは事実上それが提供する機能です。処理は行われず、代わりにクラスタ間の通信用の代替通信チャンネルを提供します。



自動フェイルオーバーの最大の課題はスプリットブレインの問題であり、この問題は2つのサイト間の接続が失われた場合に発生します。何が起るべきでしょうか？2つの異なるサイトがデータのサバイバーコピーとして自分自身を指定する必要はありませんが、1つのサイトでは、反対側のサイトが実際に失われたことと、反対側のサイトと通信できないことを区別するにはどうすればよいでしょうか。

ここでメディエーターが写真に入ります3番目のサイトに配置され、各サイトからそのサイトへの個別のネットワーク接続がある場合は、他のサイトの正常性を検証するための追加のパスが各サイトに用意されています。上の図をもう一度見て、次のシナリオを検討してください。

- 一方または両方のサイトからメディエーターに障害が発生した場合、またはメディエーターに到達できない場合はどうなりますか？
 - 2つのクラスタは、レプリケーションサービスに使用されるのと同じリンクを介して相互に通信できません。
 - データは引き続きRPO=0の保護で提供される
- サイトAに障害が発生した場合の動作
 - サイトBは、両方の通信チャンネルがダウンしたことを確認します。
 - サイトBがデータサービスをテイクオーバーするが、RPO=0ミラーリングなし
- サイトBで障害が発生した場合の動作
 - サイトAでは、両方の通信チャンネルがダウンしていることが確認されます。
 - サイトAがデータサービスをテイクオーバーするが、RPO=0ミラーリングなし

もう1つ考慮すべきシナリオがあります。データレプリケーションリンクの停止です。サイト間のレプリケーションリンクが失われた場合、RPO=0のミラーリングは明らかに不可能です。ではどうすればいいのでしょうか。

これは、優先サイトのステータスによって制御されます。SM-AS関係では、一方のサイトがもう一方のサイトのセカンダリになります。これは通常の運用には影響せず、すべてのデータアクセスは対称的ですが、レプリケーションが中断された場合は、運用を再開するためにこの関係を解除する必要があります。その結果、優先サイトはミラーリングなしで処理を継続し、レプリケーション通信がリストアされるまでセカンダリサイトはIO処理を停止します。

優先サイト

SnapMirrorのアクティブな同期の動作は対称ですが、重要な例外が1つあります（推奨サイト構成）。

SnapMirrorアクティブ同期では、一方のサイトが「ソース」で、もう一方が「デスティネーション」と見なされます。これは一方向のレプリケーション関係を意味しますが、IO動作には適用されません。レプリケーションは双方向であり、対称であり、IO応答時間はミラーの両側で同じです。

`source` 指定は、優先サイトを制御します。レプリケーションリンクが失われた場合、ソースコピー上のLUNパスは引き続きデータを提供しますが、デスティネーションコピー上のLUNパスは、レプリケーションが再確立されてSnapMirrorが同期状態に戻るまで使用できなくなります。その後、パスでデータの提供が再開されます。

ソース/デスティネーションの設定はSystemManagerで確認できます。

Relationships

Local destinations Local sources

Search Download Show/hide: Filter

Source	Destination	Policy type
▼ jfs_as1:/cg/jfsAA	jfs_as2:/cg/jfsAA	Synchronous

または、CLIで次の操作を行います。

```
Cluster2::> snapmirror show -destination-path jfs_as2:/cg/jfsAA

          Source Path: jfs_as1:/cg/jfsAA
          Destination Path: jfs_as2:/cg/jfsAA
          Relationship Type: XDP
          Relationship Group Type: consistencygroup
          SnapMirror Schedule: -
          SnapMirror Policy Type: automated-failover-duplex
          SnapMirror Policy: AutomatedFailOverDuplex
          Tries Limit: -
          Throttle (KB/sec): -
          Mirror State: Snapmirrored
          Relationship Status: InSync
```

重要なのは、ソースがcluster1のSVMであることです。前述のように、「ソース」と「デスティネーション」という用語は、レプリケートされたデータのフローを表していません。両方のサイトが書き込みを処理し、反対側のサイトにレプリケートできます。実際には、両方のクラスタがソースとデスティネーションです。1つのクラスタをソースとして指定すると、レプリケーションリンクが失われた場合に、どのクラスタが読み取り/書き込みストレージシステムとして残っているかが制御されます。

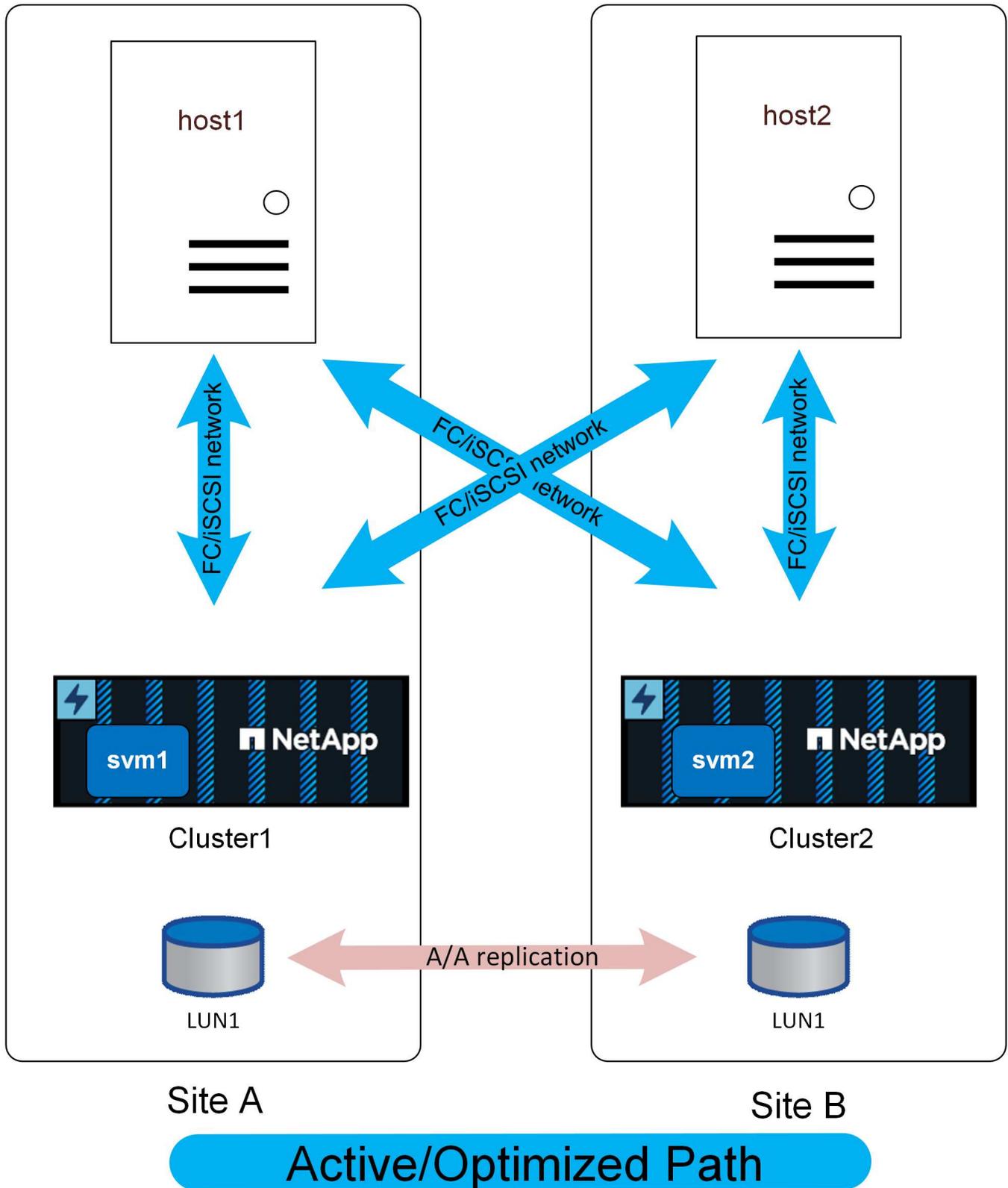
ネットワークポロジ

均一なアクセス

統一されたアクセスネットワークとは、ホストが両方のサイト（または同じサイト内の障害ドメイン）のパスにアクセスできることを意味します。

SM-ASの重要な機能の1つは、ホストがどこにあるかを認識するようにストレージシステムを設定できることです。LUNを特定のホストにマッピングするときに、LUNが特定のストレージシステムに近接しているかどうかを指定できます。

NetApp ASAシステムは、クラスタ上のすべてのパスでアクティブ/アクティブマルチパスを提供します。これはSM-AS設定にも適用されます。



アクセスが統一されている場合、IOはWANを通過します。これはフルメッシュネットワーククラスタであり、すべてのユースケースに適している場合とそうでない場合があります。

2つのサイトがファイバ接続で100m離れている場合、WANを経由する追加のレイテンシは検出されませんが、サイト間の距離が離れていると、両方のサイトで読み取りパフォーマンスが低下します。不均一なアクセ

スネットワークを使用するASAは、サイト間のレイテンシアクセスペナルティを発生させることなく、ASAのコストと機能のメリットを享受したり、ホストプロキシミティ機能を使用して両方のサイトでサイトローカルの読み取り/書き込みアクセスを許可したりすることができます。

低レイテンシ構成でSM-ASを使用するASAには、2つの興味深い利点があります。まず、I/Oは2倍のパスを使用して2倍のコントローラで処理できるため、1台のホストのパフォーマンスが実質的に2倍になります。2つ目は、単一サイト環境では、ホストへのアクセスを中断することなくストレージシステム全体が失われる可能性があるため、非常に高い可用性を提供することです。

近接設定

プロキシミティとは、特定のホストWWNまたはiSCSIイニシエータIDがローカルホストに属していることを示すクラスタ単位の構成を指します。これは、LUNアクセスを設定するための2番目のオプションの手順です。

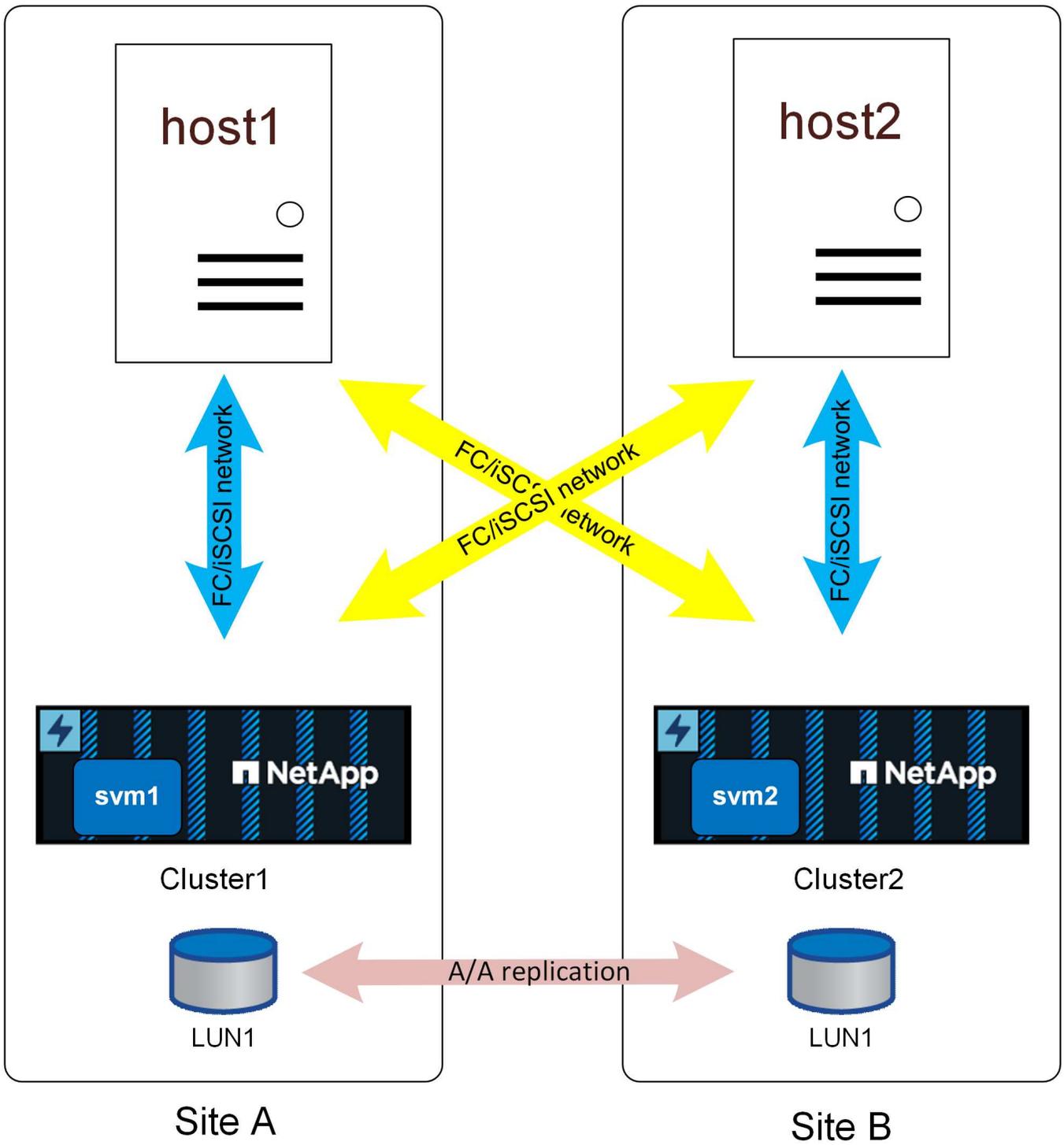
最初の手順では、通常のigroup設定を行います。各LUNは、そのLUNにアクセスする必要があるホストのWWN/iSCSI IDを含むigroupにマッピングする必要があります。これは、どのホストがLUNに_access_toを持つかを制御します。

2番目のオプション手順は、ホストプロキシミティを設定することです。これはアクセスを制御するのではなく、_priority_を制御します。

たとえば、サイトAのホストがSnapMirror Active Syncで保護されているLUNにアクセスするように設定されている場合、SANがサイト間で拡張されるため、サイトAのストレージまたはサイトBのストレージを使用してそのLUNへのパスを使用できます。

近接設定を使用しない場合、両方のストレージシステムがアクティブな最適パスをアダプタイズするため、そのホストは両方のストレージシステムを均等に使用します。SANのレイテンシやサイト間の帯域幅に制限がある場合は、この設定を解除できない可能性があります。また、通常動作中に各ホストがローカルストレージシステムへのパスを優先的に使用するように設定することもできます。これは、ホストWWN/iSCSI IDをローカルクラスタに近接ホストとして追加することで設定します。これは、CLIまたはSystemManagerで実行できます。

ホストプロキシミティが設定されている場合、パスは次のように表示されます。

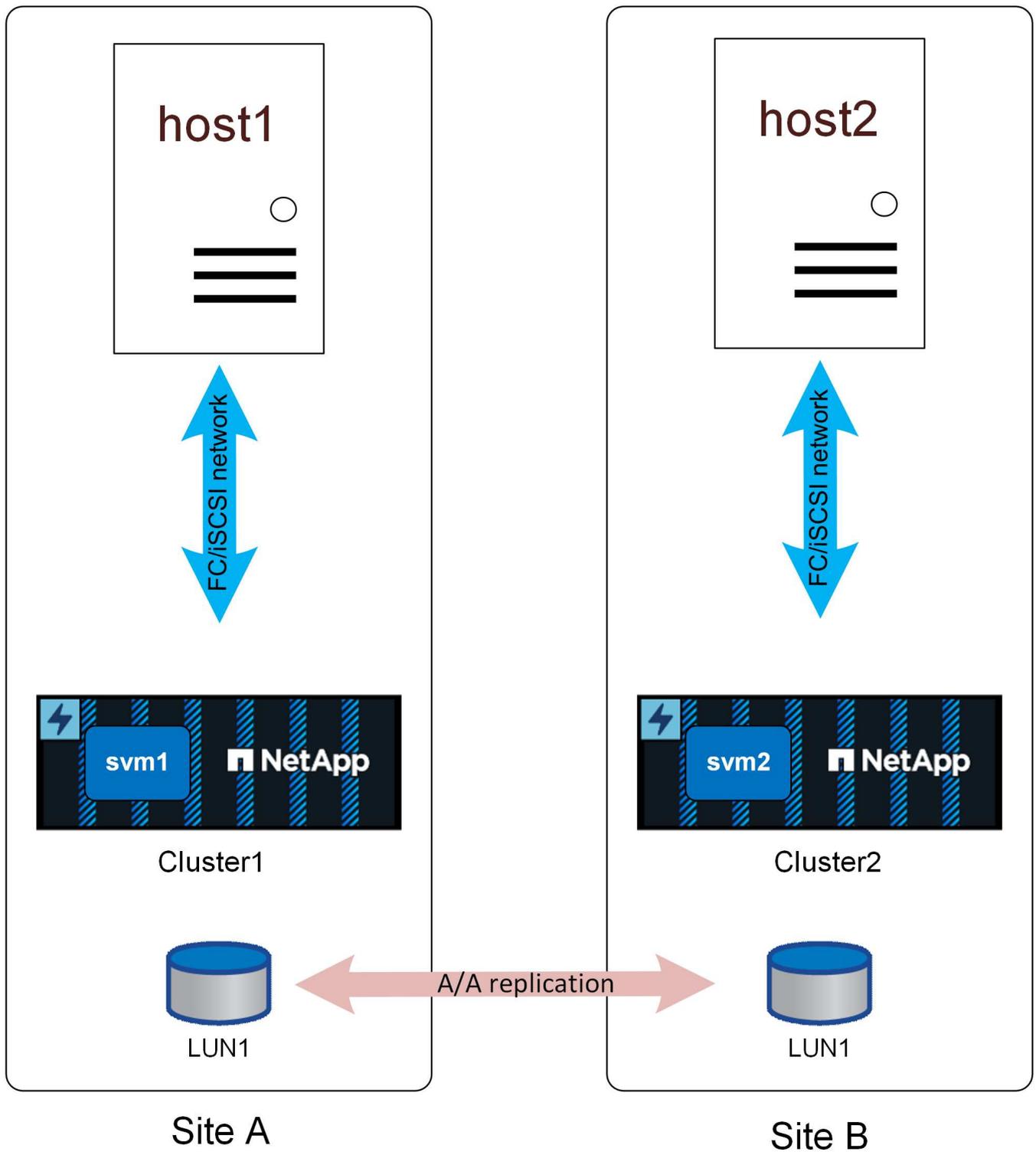


Active/Optimized Path

Active Path

不均一なアクセス

非ユニフォームアクセスネットワークとは、各ホストがローカルストレージシステム上のポートにしかアクセスできないことを意味します。SANを複数のサイト（または同じサイト内の障害ドメイン）に拡張することはできません。



Active/Optimized Path

このアプローチの主なメリットはSANのシンプルさです。SANをネットワーク経由で拡張する必要がなくなります。お客様によっては、サイト間の接続遅延が十分でない場合や、サイト間ネットワーク経由でFC SANトラフィックをトンネリングするためのインフラストラクチャが不足している場合があります。

不均一なアクセスの欠点は、レプリケーションリンクの喪失などの特定の障害シナリオで、一部のホストがストレージにアクセスできなくなることです。ローカルストレージの接続が失われると、単一のホストでのみ実行されている非クラスタデータベースなど、単一インスタンスとして実行されるアプリケーションは失敗します。データは保護されますがデータベース・サーバはアクセスできなくなりますリモートサイトで、できれば自動化されたプロセスを使用して再起動する必要があります。たとえば、VMware HAは、あるサーバでオールバスダウン状態を検出し、バスが使用可能な別のサーバでVMを再起動できます。

一方、Oracle RACなどのクラスタ化されたアプリケーションは、2つの異なるサイトで同時に利用可能なサービスを提供できます。サイトが失われても、アプリケーションサービス全体が失われるわけではありません。サバイバーサイトでは、引き続きインスタンスを使用して実行できます。

多くの場合、サイト間リンク経由でストレージにアクセスするアプリケーションによるレイテンシのオーバーヘッドは許容できません。つまり、サイトのストレージが失われると、障害が発生したサイトのサービスをシャットダウンする必要が生じるため、統一されたネットワークの可用性の向上は最小限で済みます。

この図では、わかりやすいように、ローカルクラスタを経由する冗長パスを示していません。ONTAPストレージシステム自体はHAであるため、コントローラ障害が発生してもサイト障害は発生しません。影響を受けるサイトで使用されるローカルパスが変更されるだけです。

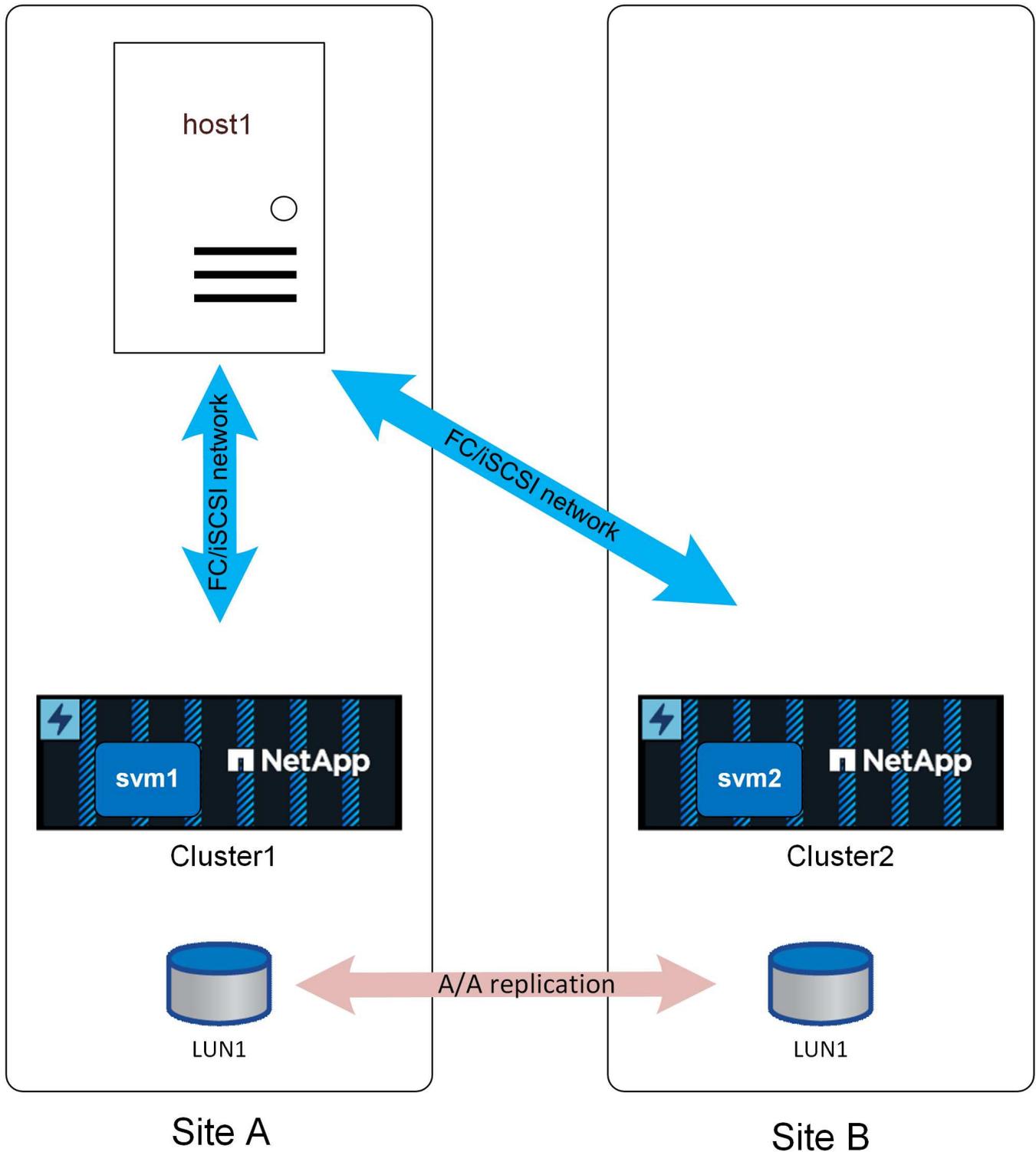
概要

SQL Serverは、いくつかの方法でSnapMirrorアクティブ同期と連携するように設定できます。適切な答えは、使用可能なネットワーク接続、RPOの要件、可用性の要件によって異なります。

SQL Serverのスタンドアロンインスタンス

ファイルレイアウトとサーバ設定のベストプラクティスは、ドキュメントで推奨されているものと同じ["ONTAP上のSQL Server"](#)です。

スタンドアロン構成では、SQL Serverを1つのサイトでのみ実行できます。おそらく["均一 \(Uniform\)"](#)アクセスが使用されます。



アクセス方法が統一されていれば、どちらかのサイトでストレージ障害が発生してもデータベースの処理は中断されません。データベース・サーバを含むサイトで完全なサイト障害が発生すると'当然'システム停止が発生します

一部のお客様は、リモートサイトで実行されているOSを、構成済みのSQL Serverセットアップで構成し、本番インスタンスと同等のビルドバージョンで更新することもできます。フェイルオーバーを実行するには、代替サイトでSQL Serverのスタンドアロンインスタンスをアクティブ化し、LUNを検出して、データベースを起動する必要があります。ストレージ側からの操作が不要なため、Windows PowerShellコマンドレットを使

用して完全なプロセスを自動化できます。

"不均一"アクセスも使用できますが、データベースにストレージへの使用可能なパスがないために、データベースサーバが配置されていたストレージシステムで障害が発生すると、データベースが停止します。これは、場合によっては許容される可能性があります。SnapMirrorのアクティブな同期では引き続きRPO=0のデータ保護が提供され、サイト障害が発生した場合でも、稼働しているコピーがアクティブになり、前述の統一されたアクセスと同じ手順で運用を再開できます。

シンプルで自動化されたフェイルオーバープロセスは、仮想化ホストを使用してより簡単に設定できます。たとえば、SQL ServerデータファイルをブートVMDKとともにセカンダリストレージに同期的にレプリケートする場合は、災害が発生したときに代替サイトで環境全体をアクティブ化できます。管理者は、サバイバーサイトでホストを手動でアクティブ化することも、VMware HAなどのサービスを使用してプロセスを自動化することもできます。

SQL Serverフェイルオーバークラスターインスタンス

SQL Serverフェイルオーバーインスタンスは、物理サーバまたは仮想サーバ上でゲストオペレーティングシステムとして実行されているWindowsフェイルオーバークラスターでホストすることもできます。このマルチホストアーキテクチャは、SQL Serverインスタンスとストレージの耐障害性を提供します。このような導入は、強化されたパフォーマンスを維持しながら堅牢なフェイルオーバープロセスを必要とする、負荷の高い環境に役立ちます。フェイルオーバークラスターのセットアップでは、ホストまたはプライマリストレージが影響を受けると、SQLサービスがセカンダリホストにフェイルオーバーされ、同時にセカンダリストレージがIOを処理できるようになります。自動化スクリプトや管理者の介入は必要ありません。

障害シナリオ

完全なSnapMirrorアクティブ同期アプリケーションアーキテクチャを計画するには、さまざまな計画的フェイルオーバーシナリオと計画外フェイルオーバーシナリオでSM-ASがどのように対応するかを理解する必要があります。

次の例では、サイトAが優先サイトとして設定されているとします。

レプリケーション接続の切断

SM-ASレプリケーションが中断されると、クラスターが反対側のサイトに変更をレプリケートできなくなるため、書き込みIOを完了できません。

サイトA（優先サイト）

優先サイトでのレプリケーションリンク障害の結果、レプリケーションリンクが本当に到達不能であると判断される前に、ONTAPがレプリケートされた書き込み処理を再試行するため、書き込みIO処理が約15秒間中断されます。15秒が経過すると、サイトAのシステムが読み取りと書き込みのIO処理を再開します。SANパスは変更されず、LUNはオンラインのままです。

サイトB

サイトBはSnapMirrorアクティブ同期優先サイトではないため、約15秒後にLUNパスが使用できなくなります。

ストレージシステムの障害

ストレージシステム障害の結果は、レプリケーションリンクが失われた場合とほぼ同じです。サバイバーサイ

トでは、IOが約15秒間停止します。その15秒が経過すると、IOは通常どおりそのサイトで再開されます。

メディアエーターの停止

メディアエーターサービスはストレージの処理を直接制御しません。クラスタ間の代替制御パスとして機能します。これは主に、スプリットブレインのリスクを伴わずにフェイルオーバーを自動化することを目的としています。通常運用時は、各クラスタがパートナーに変更内容をレプリケートするため、各クラスタはパートナークラスタがオンラインでデータを提供していることを確認できます。レプリケーションリンクに障害が発生すると、レプリケーションは停止します。

安全な自動フェイルオーバーを実現するためにメディアエーターが必要になるのは、そうしないと、双方向通信の切断がネットワークの停止によるものか実際のストレージ障害によるものかをストレージクラスタが判断できないためです。

メディアエーターは、パートナーの健全性を確認するための代替パスを各クラスタに提供します。シナリオは次のとおりです。

- クラスタがパートナーに直接接続できる場合は、レプリケーションサービスが動作しています。対処は不要です。
- 優先サイトがパートナーに直接またはメディアエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離されてLUNパスがオフラインになっているとみなされます。その後、優先サイトでRPO=0の状態が解除され、読み取りI/Oと書き込みI/Oの両方の処理が継続されます。
- 非優先サイトがパートナーに直接接続できず、メディアエーター経由で接続できる場合、そのサイトのパスはオフラインになり、レプリケーション接続が戻るまで待機します。
- 優先されないサイトがパートナーに直接、または動作中のメディアエーターを介してアクセスできない場合、パートナーが実際に使用できないか分離され、LUNパスがオフラインになったとみなされます。優先されないサイトは、RPO=0状態の解放に進み、読み取りI/Oと書き込みI/Oの両方の処理を続行します。レプリケーションソースの役割を引き継ぎ、新しい優先サイトになります。

メディアエーターが完全に使用できない場合：

- 非優先サイトまたはストレージシステムの障害など、何らかの理由でレプリケーションサービスに障害が発生すると、優先サイトでRPO=0状態が解放され、読み取りおよび書き込みIO処理が再開されます。非優先サイトのパスがオフラインになります。
- 優先サイトに障害が発生すると、非優先サイトでは、反対側のサイトが本当にオフラインであることを確認できず、そのため非優先サイトがサービスを再開しても安全ではないため、システムが停止します。

サービスのリストア

サイト間の接続のリストアや障害が発生したシステムの電源投入などの障害が解決されると、SnapMirrorのアクティブな同期エンドポイントは、障害のあるレプリケーション関係の存在を自動的に検出してRPO=0状態に戻します。同期レプリケーションが再確立されると、障害が発生したパスは再びオンラインになります。

多くの場合、クラスタ化されたアプリケーションは障害が発生したパスの復帰を自動的に検出し、それらのアプリケーションもオンラインに戻ります。また、ホストレベルのSANスキャンが必要な場合や、アプリケーションを手動でオンラインに戻す必要がある場合もあります。それはアプリケーションとそれがどのように構成されているかによって異なり、一般的にそのようなタスクは簡単に自動化することができます。ONTAP自体は自己回復型であり、RPO=0のストレージ処理を再開するためにユーザの介入は不要です。

手動フェイルオーバー

優先サイトを変更するには、簡単な操作が必要です。クラスタ間でレプリケーション動作の権限が切り替わるため、IOは1~2秒間停止しますが、それ以外の場合はIOには影響しません。

データセキュリティ

SQL Serverデータベース環境の保護は、データベース自体の管理にとどまらない多次元的な取り組みです。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。



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、[を参照してください。](#) **"ONTAPのドキュメント"**

スナップショットの改ざん防止

ONTAP 9.12.1以降では、スナップショット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は、ファブリックプールを使用して階層化できません。

改ざん防止スナップショットの詳細については、["ONTAPのドキュメント"](#)



新しいASAプラットフォームで、`_vault_policy`を使用してSnapshotをリモートクラスタにレプリケートし、デスティネーションをロックしてSnapshotの改ざんを防止します。

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スナップショットが含まれます。このスナップショットでは、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、"[ONTAPのドキュメント](#)"

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、[を参照してください。](#) "[ONTAPのドキュメント](#)"

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、[を参照してください。](#) "[ONTAPのドキュメント](#)"

多要素認証 (MFA)

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。

多要素認証 (MFA) の詳細については、"[ONTAPのドキュメント](#)"

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。

API RBACの詳細については、"[ONTAPのドキュメント](#)"

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、"[ONTAPのドキュメント](#)"

著作権に関する情報

Copyright © 2026 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。