



Sitecore CMS 6.5

SQL Server 用 DMS

パフォーマンス チューニング ガイド

Sitecore DMS のパフォーマンスを最適化するためのシステム管理者向けガイド

この文書に含まれる情報は、発行日の時点で議論されていた問題に対する Sitecore Corporation の現在の見解であり、いつでも予告なしに変更することがあります。この文書およびその内容は、いかなる種類の保証もなく、現状有姿で提供され、Sitecore 側の提供や義務として解釈されるべきものではなく、Sitecore は提供する情報について、その正確性を保証するものではありません。Sitecore は、明示または暗黙を問わず、本ドキュメントについて一切の責任を負いません。

この文書に記載されている他社の製品に関する記述は、読者の便宜を図るためにのみ提供されるものです。そうした他社製品に対する言及は、Sitecore による同製品に対する支持または支援として解釈されるものではありません。Sitecore は他社製品に関する記述の正確性を保証できかねます。また、他社製品も時間の経過とともに変わる可能性があります。そうした製品の説明は理解を助けるための簡単な概要として記述したものであり、詳細な解説を意図したものではありません。そうした製品の正式な説明については、各メーカーにお問い合わせください。

すべての商標はそれぞれ各社の所有物です。

©2012 Sitecore Corporation. All rights reserved.

目次

第 1 章	イントロダクション	4
1.1	必要条件 - データベース互換性レベル	5
1.2	SQL Server インデックスの断片化レベル	7
1.3	ロボットトラフィックの自動検出を有効化する	9
1.4	SQL Server メンテナンス プラン	11
第 2 章	データベースのプロパティ	14
2.1	[復旧モデル] を [単純] に設定する	15
2.2	AutoClose プロパティを False に設定する	17
2.3	Auto Shrink プロパティを False に設定する	19
2.4	初期サイズ値を設定してからデータを挿入する	21
2.5	Autogrowth プロパティを設定してからデータを挿入する	23
2.6	接続文字列パラメーター	26
第 3 章	サーバーに関する推奨事項	28
3.1	DMS のサーバーに関する推奨事項	29
第 4 章	Analytics 設定パラメーター	32
4.1	パラメーターの動作	33
4.2	チューニング シナリオ	35

第 1 章

イントロダクション

DMS チューニング ガイドは、DMS 2.0以降のパフォーマンスを改善するために記述されたものです。このガイドは、タスクのセットとして構成されています。各タスクは重要度の高いものから順に並んでいます。

各タスクは、症状の説明、推奨されるチューニング、補足分析から成ります。

この章には次のセクションがあります。

- 必要条件 - データベース互換性レベル
- SQL Server インデックスの断片化レベル
- ロボットトラフィックの自動検出を有効化する
- SQL Server メンテナンス プラン

1.1 必要条件 - データベース互換性レベル

互換性レベルは、SQL 構文やクエリの解析には影響を与えますが、パフォーマンスには影響を与えないはずですが。互換性レベルを SQL Server 2008(100) という値に設定すると、新しい T-SQL 機能を利用できるようになります。この機能は、『DMS パフォーマンス チューニング ガイド』に登場する多数のスクリプトやコマンドで使用されています。

1.1.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

1.1.2 症状

- DMS チューニングに必要な T-SQL スクリプトを実行できない。

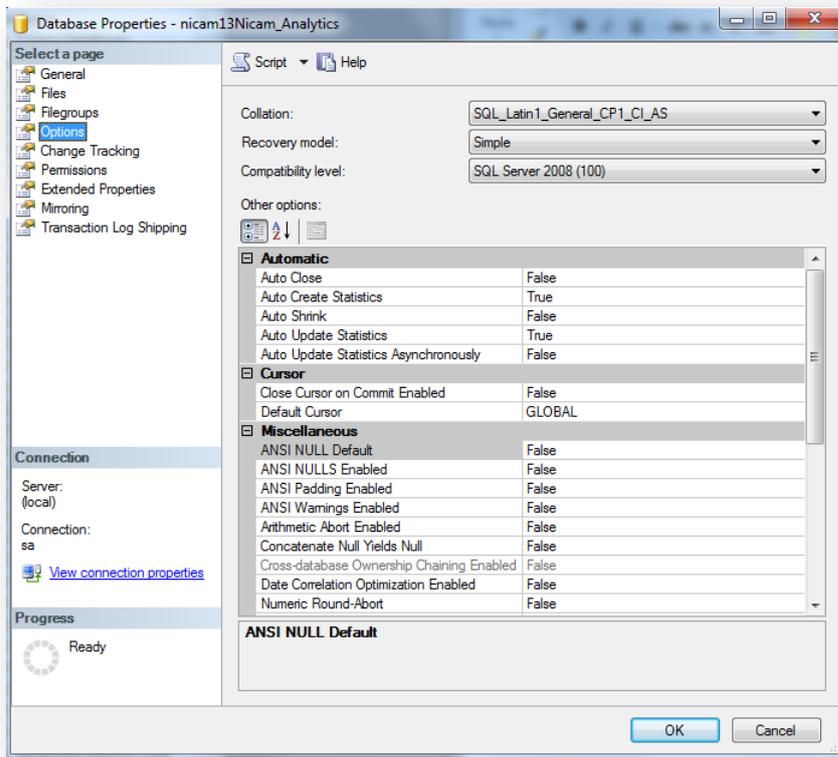
1.1.3 データベース互換性レベルの確認

データベースの [互換性レベル] プロパティの値を確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、[互換性レベル] プロパティを確認します。

1.1.4 結果の確認

出力は、次のようになります。



1.1.5 Sitecore の推奨

Sitecore では、[互換性レベル] プロパティを SQL Server 2008(100) に設定することを推奨しています。

1.1.6 解決方法

[互換性レベル] プロパティを設定するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、[互換性レベル] が SQL Server 2008(100) に設定されていることを確認します。

1.2 SQL Server インデックスの断片化レベル

インデックスは時間がたつにつれ、不連続なデータの挿入や削除が悪い効果をもたらし、断片化を発生させます。これは、処理量が多い DMS データベースではほんの数日で起きる場合があります。少量の断片化では通常、パフォーマンスを損ないません。しかし断片化の割合が増えるにつれ、パフォーマンスは大幅に低下します。

1.2.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

1.2.2 症状

- CPU 使用率が劇的に増加する。
- クエリのパフォーマンスが低下する。
- データベース書き込みのパフォーマンスが低下する。
- データベース サーバーへの接続が切断される。
- レポートの遅延または未応答が発生する。

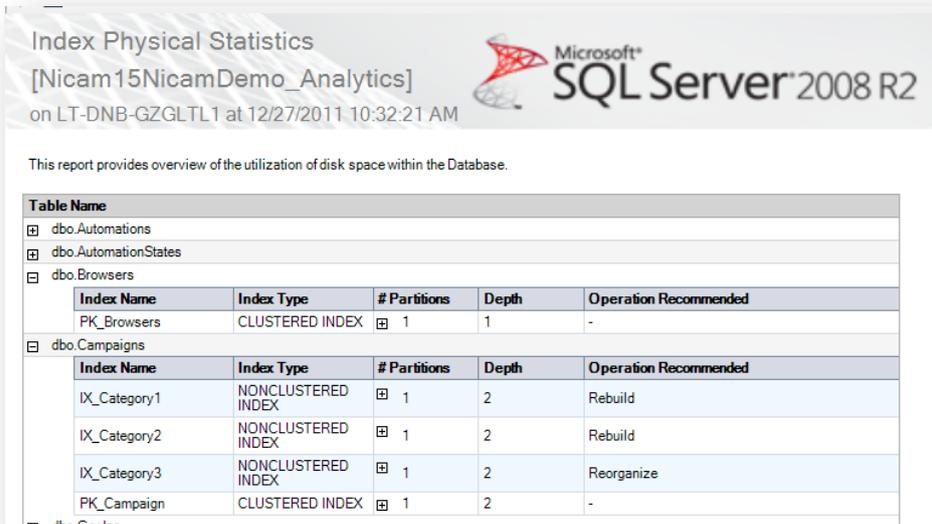
1.2.3 インデックスの断片化が発生しているかどうかの確認

インデックスの断片化の割合を確認し、次の手順に従って、DMS データベースに対して [インデックス物理統計標準レポート] を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [データベースのプロパティ] ウィンドウで [オプション] ページを選択し、[互換性レベル] が SQL Server 2008(100) に設定されていることを確認します。
4. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[レポート]、[標準レポート]、[インデックス物理統計] の順に選択します。
5. SQL Server Management Studio によってレポートが生成され、[テーブル名]、[インデックス名]、[インデックス タイプ]、[パーティション番号]、および [推奨操作] についての情報が表示されます。

1.2.4 結果の確認

出力は、次のようになります。



Index Physical Statistics
[Nicom15NicomDemo_Analytics]
on LT-DNB-GZGLTL1 at 12/27/2011 10:32:21 AM

This report provides overview of the utilization of disk space within the Database.

Table Name	Index Name	Index Type	# Partitions	Depth	Operation Recommended
dbo.AutomationStates	PK_Browsers	CLUSTERED INDEX	1	1	-
dbo.Campaigns	IX_Category1	NONCLUSTERED INDEX	1	2	Rebuild
	IX_Category2	NONCLUSTERED INDEX	1	2	Rebuild
	IX_Category3	NONCLUSTERED INDEX	1	2	Reorganize
	PK_Campaign	CLUSTERED INDEX	1	2	-

このレポートの [推奨操作] フィールドの値は重要です。このフィールドの値が "Rebuild" の場合、インデックスが断片化されていることを意味します。

[パーティション番号] フィールドを展開すると、当該インデックスの断片化の度合 (%) を確認できます。

Index Name	Index Type	# Partitions	Depth	Operation Recommended		
IX_Category1	NONCLUSTERED INDEX	1	2	Rebuild		
		Partition No.	Avg. Fragmentation (%)	# Fragments	Avg. Pages Per Fragment	# Pages
		1	43	4	2	7

1.2.5 Sitecore の推奨

Sitecore では、インデックスの断片化を 10% 未満に抑えるように推奨しています。

1.2.6 解決方法

DMS データベースのインデックスを最適化するには、最適化メンテナンス プランを実行する必要があります。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、[管理/メンテナンス プラン] フォルダを展開します。
3. [インデックス最適化メンテナンス プラン*] を右クリックし、[実行] をクリックします。

*このメンテナンス プランが存在しない場合は、「SQL Server メンテナンス プランが存在しているかどうかの確認」タスクを参照してください。

1.3 ロボット トラフィックの自動検出を有効化する

ロボット トラフィック (Web クローラー トラフィック) は、Google などの検索エンジンが Web サイトのインデックスを生成するときに発生します。Web クローラーによって生成されるトラフィックは膨大な量になることがあり、DMS データベースが短期間で急速に成長する可能性があります。

DMS は、こうしたロボット トラフィックを設定によって自動的にブロックする機能を備えています。このタスクでは、この機能を有効にするように構成が設定されていることを確認します。

1.3.1 必要なスキル

- Sitecore 設定ファイルの実用的な知識。

1.3.2 症状

- DMS アナリティクス データベースのサイズが劇的に増加する。

1.3.3 Analytics.AutoDetectBots 設定の確認

Analytics.AutoDetectBots 設定が有効になっていることを確認するには、次の手順を実行します。

- <webroot>/App_Config/Include ディレクトリに移動し、任意のエディターで Sitecore.Analytics.config ファイルを開きます。

1.3.4 結果の確認

出力は、次のようになります。

```
<configurationxmlns:patch="http://www.sitecore.net/xmlconfig/">
  <sitecore>
    <settings>
      <!-- ANALYTICS AUTO DETECT BOTS
           Enable auto detection of bots.
           Default: true
      -->
      <settingname="Analytics.AutoDetectBots" value="true" />
    
```

Analytics.AutoDetectBots パラメーターによって、DMS によるロボット トラフィックの自動検出が有効になります。これが有効になっていると、ロボット トラフィックは、データベースによって捕捉される前にブロックされます。

1.3.5 Sitecore の推奨

Sitecore では、DMS の場合、Analytics.AutoDetectBots パラメーターを true に設定して不要なロボット トラフィックをブロックすることを推奨しています。これにより、DMS データベースの不要な成長を抑えることができます。

1.3.6 解決方法

Analytics.AutoDetectBots 設定を有効にするには、次の手順を実行します。

1. <webroot>/App_Config/Include ディレクトリに移動し、任意のエディターで Sitecore.Analytics.config ファイルを開きます。
2. Analytics.AutoDetectBots パラメーターを true に設定します。

1.4 SQL Server メンテナンス プラン

メンテナンス プランでは、スケジュールに従ってタスクの自動セットを実行することにより、データベースの手動メンテナンスの必要性がなくなります。このプランは、データベースの定期的なチェックとメンテナンスを行い、確実にデータベースを最適な状態にします。

1.4.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

1.4.2 症状

- インデックスが断片化されていると、検索所要時間が長くなり、タイムアウトが発生する。

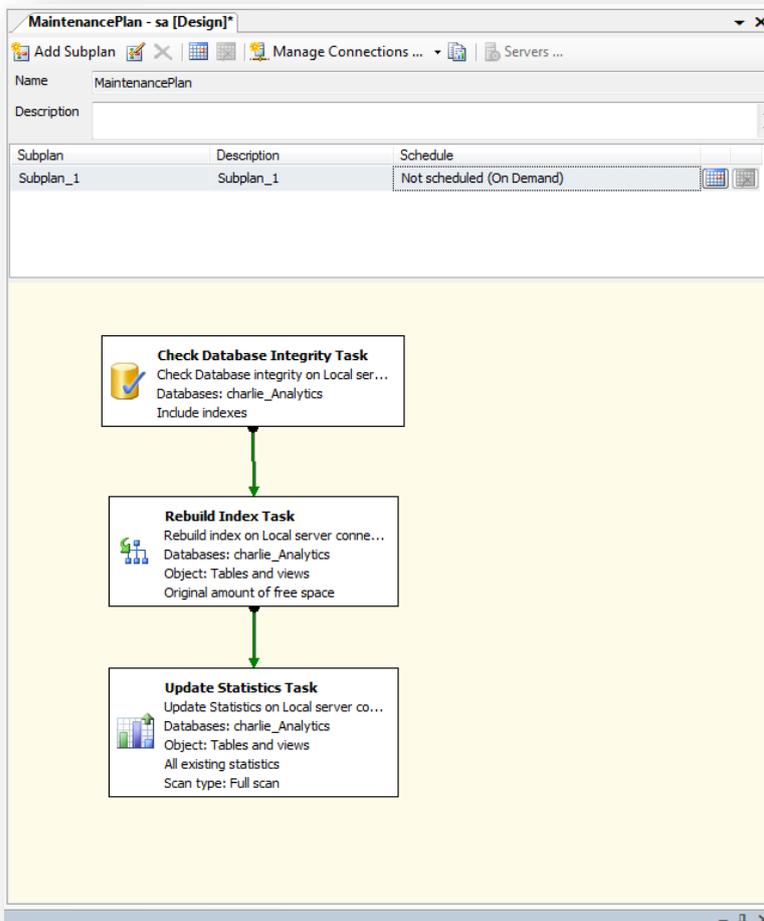
1.4.3 SQL Server メンテナンス プランが存在しているかどうかの確認

SQL Server メンテナンス プランが存在しているかどうか、そのプランが Sitecore のベスト プラクティスに従っているかどうかを確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、[管理/メンテナンス プラン] フォルダーを展開します。
3. メンテナンス プランが存在する場合は、そのプランをダブルクリックして設定を確認します。このプランは、このタスクの「結果」セクションで使用します。

1.4.4 結果の確認

出力は、次のようになります。



メンテナンス プランの内容は次のとおりです。

- データベースの整合性確認タスク
- インデックスの再構築タスク
- 統計情報の更新タスク

1.4.5 Sitecore の推奨

Sitecore では、DMS データベースの SQL Server メンテナンス プランを用意しておくことを推奨しています。メンテナンス プランには、データベースの整合性確認タスク、インデックスの再構築タスク、統計情報の更新タスクが含まれていなければなりません。

1.4.6 解決方法

SQL Server Management Studio には、メンテナンス プランの作成を簡素化する IDE が用意されています。インデックスを最適化するメンテナンス プランを作成するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、[管理] フォルダーを展開します。
3. [メンテナンス プラン] フォルダーを右クリックし、[新しいメンテナンス プラン] を選択します。
4. メンテナンス プランに、「DMS インデックスの最適化」などわかりやすい名前を付けます。
5. ツールボックスから [データベースの整合性確認タスク]、[インデックスの再構築タスク]、[統計情報の更新タスク] をドラッグ アンド ドロップし、この順序で縦方向に配置します。
6. 矢印を 1 つのボックスから他のボックスにドラッグしてタスク間を接続し、次のように接続されるようにします。
[データベースの整合性確認タスク] -> [インデックスの再構築タスク] -> [統計情報の更新タスク]。
7. [データベースの整合性確認タスク] を右クリックし、[編集] を選択します。
8. [接続] と DMS データベースを選択し、[OK] をクリックします。
9. [インデックスの再構築タスク] を右クリックし、[編集] を選択します。
10. [接続] と DMS データベースを選択し、[OK] をクリックします。

注:

MSSQL Server Enterprise Edition 以上のエディションを実行している場合は、[高度な設定] の [インデックスの再構築中もインデックスをオンラインのまま維持する] を有効にすることをお勧めします。

11. [データベースの整合性確認タスク] を右クリックし、[編集] を選択します。
12. [接続] と DMS データベースを選択し、[オブジェクト] を [テーブルとビュー] に、[更新] を [すべての既存の統計] に、[スキャンの種類] を [フル スキャン] に設定して、[OK] をクリックします。
13. [スケジュール] の隣のカレンダー アイコンをクリックし、日次で実行するようにスケジュールを設定します。
14. 変更を保存します。

第 2 章

データベースのプロパティ

この章では、DMS データベースのパフォーマンスに影響を与えるいくつかの重要なデータベース プロパティについて説明します。

この章には次のセクションがあります。

- [復旧モデル] を [単純] に設定する
- AutoClose プロパティを False に設定する
- Auto Shrink プロパティを False に設定する
- 初期サイズ値を設定してからデータを挿入する
- Autogrowth プロパティを設定してからデータを挿入する
- 接続文字列パラメーター

2.1 [復旧モデル] を [単純] に設定する

[単純復旧モデル] 設定を選択すると、SQL Server がトランザクション ログに記録する情報量が最小限に抑えられます。SQL Server は基本的に、トランザクション ログが最大容量の 70% に達したとき、またはトランザクション ログのアクティブ部分が、[復旧インターバル] サーバー レベル設定に指定された時間内に復旧できるサイズを超えたときは常に、トランザクション ログの末尾を切り捨てます。

[復旧モデル] を [単純] に設定すると、[フル] または [大容量ログ] に設定したときに比べてオーバーヘッドが最小限に抑えられます。これは、DMS データベースに必要なパフォーマンス要件を維持する上で非常に重要です。

注:

高可用性 (HA) 設定の場合は、[復旧モデル] を [フル] に設定することをお勧めします。詳細については、『[Sitecore スケーリング ガイド](#)』を参照してください。

2.1.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

2.1.2 症状

- データベースの復旧に長時間を要する。

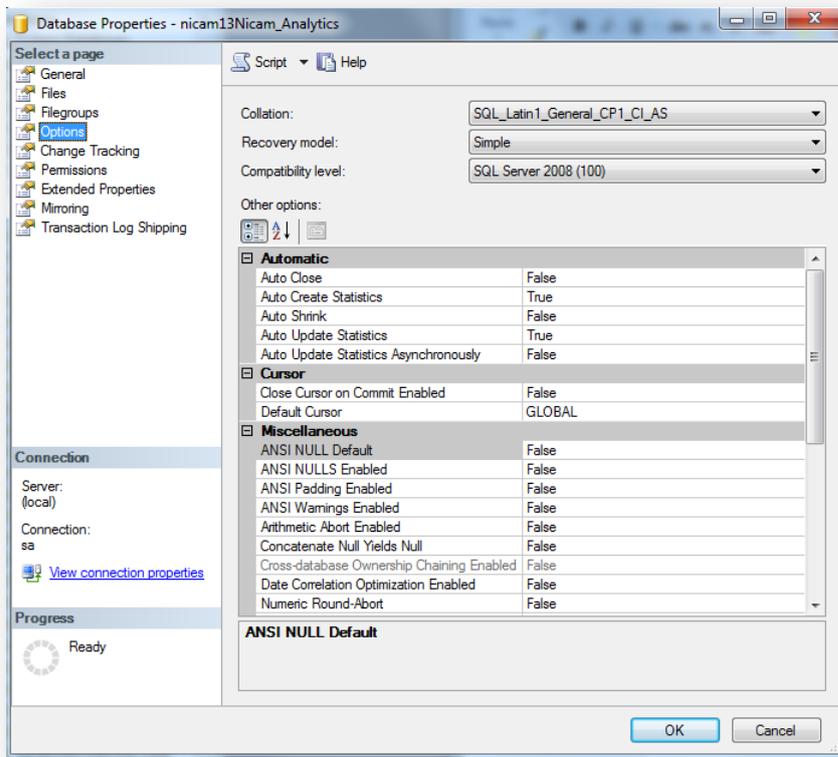
2.1.3 復旧モデルの確認

[復旧モデル] プロパティの値を確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、[復旧モデル] プロパティを確認します。

2.1.4 結果の確認

出力は、次のようになります。



2.1.5 Sitecore の推奨

Sitecore では、[復旧モデル] プロパティを [単純] に設定することを推奨しています。

2.1.6 解決方法

[復旧モデル] プロパティを設定するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、[復旧モデル] が [単純] に設定されていることを確認します。

2.2 AutoClose プロパティを False に設定する

SQL Server がデータベースをオープンすると、オープン状態を維持するため各種リソースが割り当てられます。ロック、バッファ、セキュリティ トークンなどに使用されるメモリがすべて割り当てられます。

この操作には時間がかかります。Auto Close プロパティは、これらのリソースの管理方法を指示します。このプロパティが true または ON に設定されている場合、最後の接続がクローズされると、これらのリソースが解放されます。これは適切な動作のように思われますが、極めて短い時間 (1/10 秒以下) で次の新しい接続要求を受信すると、すべてのリソースをすぐに割り当て直す必要があります。Auto Close プロパティを false または OFF に設定すると、このような動作を抑制できます。

2.2.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

2.2.2 症状

- データベースの接続に長時間を要する。

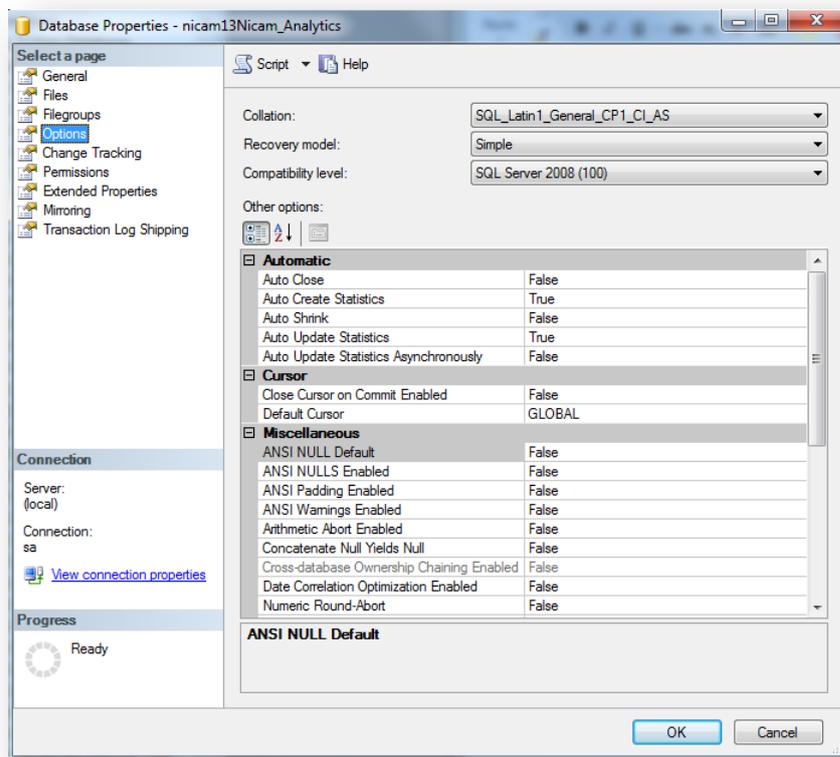
2.2.3 Auto Close プロパティの値の確認

Auto Close プロパティの値を確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、Auto Close プロパティを確認します。

2.2.4 結果の確認

出力は、次のようになります。



2.2.5 Sitecore の推奨

Sitecore では、Auto Close プロパティを False に設定するよう推奨しています。

2.2.6 解決方法

Auto Close プロパティを設定するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、Auto Close プロパティが False に設定されていることを確認します。

2.3 Auto Shrink プロパティを False に設定する

Auto Shrink プロパティには、次のような欠点があります。

- 呼び出し時に大量のリソースを消費する。
- 呼び出しの時期を制御できない。

Auto Shrink と Auto Growth を組み合わせると、データベースの拡大および縮小が頻繁に発生するスパイラル状態に陥り、他のデータベース タスクが貴重なリソースを奪われたり、断片化の問題が発生したりする可能性があります。データベースまたはファイルを縮小する必要がある場合は、スクリプト、コマンド、あるいはスケジュールされたメンテナンスプランを使用するようにしてください。Auto Shrink プロパティを false または OFF に設定すると、自動縮小機能が無効になります。

2.3.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

2.3.2 症状

- パフォーマンスが低下する。

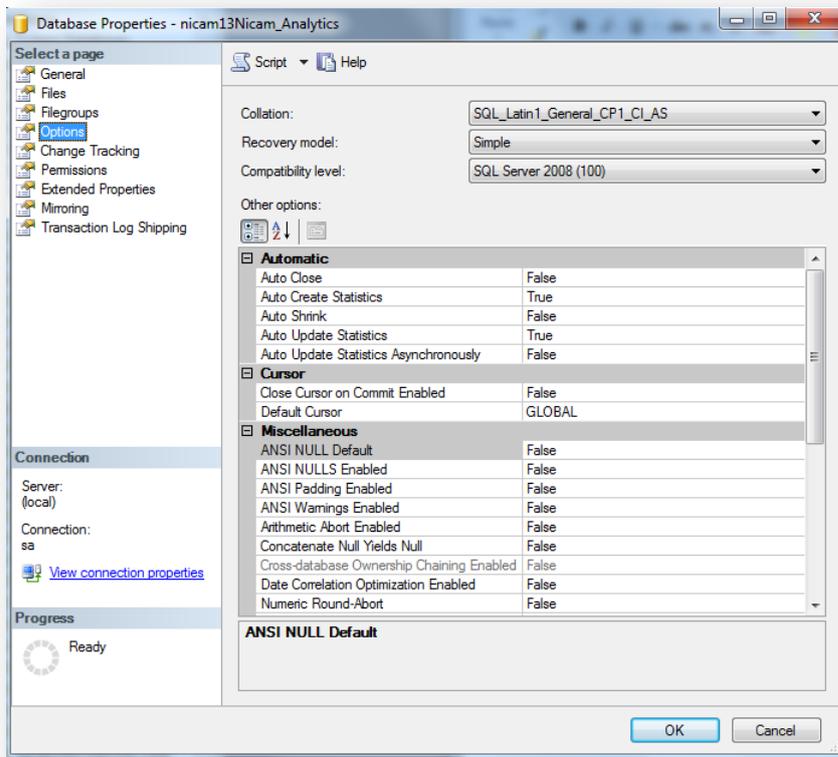
2.3.3 Auto Shrink プロパティの値の確認

Auto Shrink プロパティの値を確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、Auto Shrink プロパティを確認します。

2.3.4 結果の確認

出力は、次のようになります。



2.3.5 Sitecore の推奨

Sitecore では、Auto Shrink プロパティを False に設定するよう推奨しています。

2.3.6 解決方法

Auto Shrink プロパティを設定するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [オプション] ページを選択し、Auto Shrink プロパティが False に設定されていることを確認します。

2.4 初期サイズ値を設定してからデータを挿入する

データベースの [初期サイズ] プロパティの値を 3 ～ 6 か月分のデータを収容できる値に設定しておくことは重要です。これにより、自動拡張が発生する頻度を抑えることができます。

自動拡張操作は、実行に使用するリソースの点で高コストというだけでなく、断片化の問題を引き起こす可能性もあります。

2.4.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

2.4.2 症状

- パフォーマンスが低下する。
- 頻繁な自動拡張コマンドによりリソースが過剰に消費される。
- データベース インデックスがすぐに断片化する。

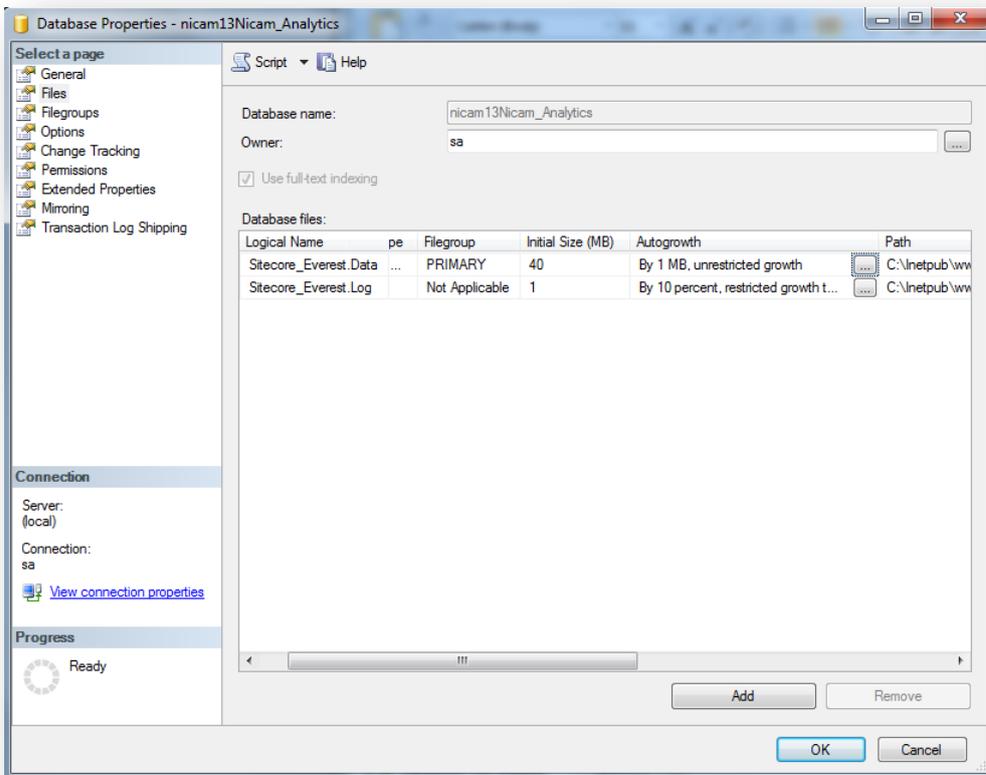
2.4.3 初期サイズ値の確認

初期サイズ値を確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [ファイル] ページを選択し、初期サイズの値を確認します。

2.4.4 結果の確認

出力は、次のようになります。



2.4.5 Sitecore の推奨

Sitecore では、データベースの [初期サイズ] プロパティの値を 3 ~ 6 か月分のデータを収容できる値に設定しておくことを推奨しています。

2.4.6 解決方法

初期サイズの値を設定するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [ファイル] ページを選択し、[初期サイズ] プロパティの値を 3 ~ 6 か月分のデータ格納容量に設定します。

2.5 Autogrowth プロパティを設定してからデータを挿入する

Autogrowth プロパティを間違えて設定すると、パフォーマンスが低下することがあります。Autogrowth プロパティの設定値が低すぎると、より大きなスペースを必要とするトランザクションが発生した場合、そのトランザクションは、拡張操作が行われるまで処理を完了できません。また、データベースの拡張操作を頻繁に行うと、断片化の問題が発生します。

設定時に指定する正確な値や、拡張サイズを拡張率 (パーセンテージ) で指定するか具体的な MB 値で指定するかは、お使いの環境のさまざまな要因によって異なります。大体の目安として、自動拡張の値をファイル サイズの 8 分の 1 くらいに設定してみてください。

2.5.1 必要なスキル

- SQL Server 2008 Management Studio の実用的な知識。

2.5.2 症状

- パフォーマンスが低下する。

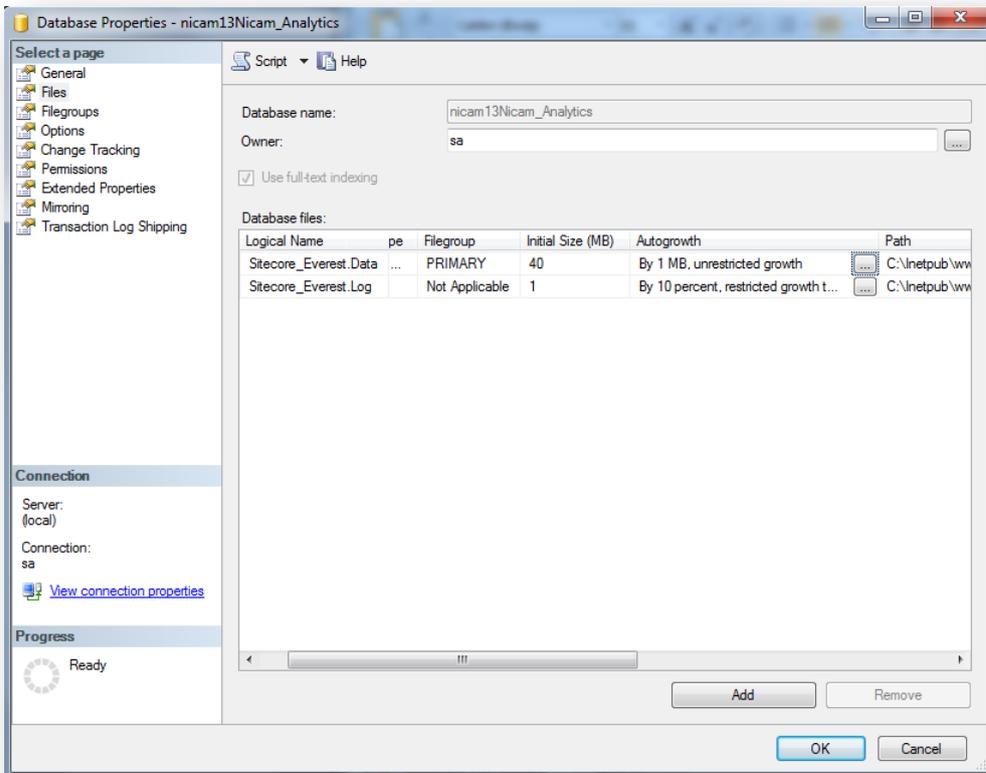
2.5.3 Autogrowth プロパティの値の確認

Autogrowth プロパティの値を確認するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [ファイル] ページを選択し、Autogrowth プロパティを確認します。

2.5.4 結果の確認

出力は、次のようになります。



2.5.5 Sitecore の推奨

Sitecore では、Autogrowth プロパティを有効にし、拡張率を 10 ~ 15% に設定するよう推奨しています。

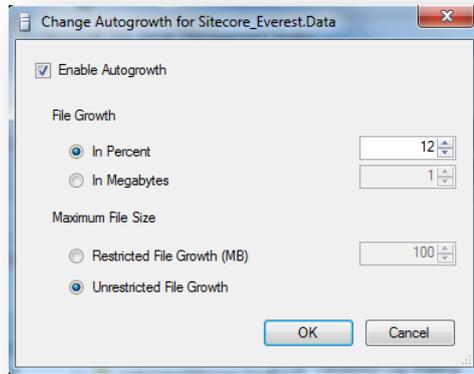
2.5.6 解決方法

Autogrowth プロパティを有効にして、拡張率を設定するには、次の手順を実行します。

1. SQL Server Management Studio を起動します。
2. オブジェクト エクスプローラーで、DMS データベースを右クリックし、[プロパティ] を選択します。
3. [ファイル] ページを選択し、Autogrowth プロパティで  ボタンをクリックして [自動拡張の変更] ダイアログボックスを開きます。
4. [自動拡張を有効にする] チェックボックスをオンにします。
5. [ファイルの拡張] セクションで、[パーセント] オプションを選択し、10 ~ 15% (データベース サイズの約 8 分の 1) の値を設定します。

6. [最大ファイル サイズ] セクションで、[無制限ファイル拡張] オプションを選択します。

ダイアログボックスは、次のようになります。



2.5.7 注とコメント

高いトラフィック負荷が発生しているか、発生することが予想されるサイトでは、DMS データベースに割り当てる初期スペース量をできるだけ大きな値に設定する必要があります。これにより、自動拡張が発生する頻度を抑えることができます。

2.6 接続文字列パラメーター

アナリティクス接続文字列 (<webroot>/App_Config/ConnectionString.config に指定されている) には、接続文字列の保持方法と接続文字列のタイムアウト時期を制御するパラメーターが含まれています。これらのパラメーターを正しく設定することで、Sitecore CMS と Sitecore DMS アナリティクス データベース間の通信のパフォーマンスを改善できます。

次の 2 つの接続文字列パラメーターを確認する必要があります。

- 接続タイムアウト
 - 接続タイムアウトは、ログ ファイルにアナリティクス データベースへの接続に関連する TCP エラーが書き込まれている場合のみ設定する必要があります。
- 最少プール サイズ

2.6.1 必要なスキル

- Sitecore 設定ファイルの実用的な知識。
- Sitecore 設定ファイルの解析に関する実用的な知識。

2.6.2 症状

- TCP タイムアウト エラーが発生する。
- DMS データベース サーバーへの接続を作成するのに時間がかかり過ぎる。

2.6.3 アナリティクス接続文字列パラメーターの確認

アナリティクス接続文字列のパラメーターを確認するには、次の手順を実行します。

1. <webroot>/App_Config ディレクトリに移動し、任意のエディターで ConnectionString.config ファイルを開きます。

2.6.4 結果の確認

出力は、次のようになります。

```
<?xmlversion="1.0" encoding="utf-8"?>
<connectionStrings>
<!--
  Sitecore connection strings.
  All database connections for Sitecore are configured here.
-->
```

```
<addname="core" connectionString="user id=sa;password=xxxx;Data Source=(local);Database=CMS_C
ore"/>
<addname="master" connectionString="user id=sa;password=xxxx;Data Source=(local);Database=CMS
_Master"/>
<addname="web" connectionString="user id=sa;password=xxxx;Data Source=(local);Database=CMS_W
eb"/>
<addname="analytics"
connectionString="user id=sa;password=xxxx;Data Source=(local);Database=DMS_Analytics;Connectio
n Timeout=120;Min Pool Size=5"/>
</connectionStrings>
```

Connection Timeout パラメーターはサーバーへの接続待ち時間 (秒) です。この時間を超えると、接続処理が終了し、エラーが生成されます。デフォルト値は 15 秒です。

Min Pool Size パラメーターは、プールに登録される許容最少接続数を示します。デフォルト値は 0 で、すべての接続がクローズされるとプールが空になることを意味します。0 より大きな値を設定すると、使用可能な接続がプール内に常に存在することが保証されます。これにより、すべての接続がクローズされるたびに新しい接続を再インスタンス化せずに済みます。

2.6.5 Sitecore の推奨

Sitecore では、次のように接続文字列パラメーターを設定することを推奨しています。

- Connection Timeout=120 (ログ ファイルに TCP エラーが出現している場合のみ。それ以外の場合は、Connection Timeout パラメーターを削除してもかまわない。削除すると、デフォルト値である 15 秒が使用される)
- Min Pool Size=5

2.6.6 解決方法

Connection String パラメーターを設定するには、次の手順を実行します。

1. <webroot>/App_Config ディレクトリに移動し、任意のエディターで ConnectionString.config ファイルを開きます。
2. アナリティクス接続文字列を編集して、設定ファイルを保存します。

第 3 章

サーバーに関する推奨事項

DMS のハードウェアサイジングを行う際には、膨大なデータとトラフィックが Web サイトで処理される可能性があることを考慮する必要があります。以下に示すサーバーに関する推奨事項は、SQL Server のエンタープライズレベルの配備を想定したものです。

この章には次のセクションがあります。

- DMS のサーバーに関する推奨事項

3.1 DMS のサーバーに関する推奨事項

以下に示すチェック表は、DMS アナリティクス データベースをホストするサーバーの設定が最適なパフォーマンスを実現するのに十分なレベルであるかどうかを確認するために使用します。チェック表では、サーバーのパフォーマンス レベルを記録するために、最低限のレベルから始まる評価システムを使用しています。評価は各セクションに分かれており、各セクションにそれぞれ固有のスコアを付けます。スコアが 0 のセクションは、そのセクションが Sitecore の最低推奨レベルを満たしていることを示します。スコアが 0 より大きい場合は、データベースが高レベルのパフォーマンスを実現できることを示します。スコアが 0 より小さい場合は、警告を意味し、Sitecore の最低推奨レベルを満たすためには、サーバーに対するリソースの追加または変更が必要であることを示します。

この評価システムは、SDN の DMS サーバー要件に挙げた最小要件に基づいています。

Sitecore では、DMS アナリティクス データベースを稼働する場合は、次の最小サーバー要件を満たすように推奨しています。

- 専用の物理サーバー
 - 1 クアッド コア以上の CPU
 - 8 GB 以上の RAM
 - OS 用のディスク 2 基
 - データベース用のディスク 2 基 (データベース用とログ用に 1 基ずつ)
 - TempDb 用のディスク 1 基
 - MS SQL サーバー要件に準拠した RAID ディスク設定
- Windows Server 2003、2008、または 2008 R2 (強く推奨)
 - 64 ビット版を推奨
- Windows SQL Server 2008、または 2008 R2 (強く推奨)
 - 64 ビット版を推奨

3.1.1 必要なスキル

- システム インフラストラクチャに関する実用的な知識。

3.1.2 症状

- CPU 負荷の高い状態が続く。
- メモリが過剰に消費される。
- I/O 待ち時間が長く、ディスク使用量が多い。

- パフォーマンスが低下する。
- レポート タイムアウトが発生する。

3.1.3 サーバー構成の評価

サーバー構成の評価は大きく 2 つのカテゴリに分かれており、各カテゴリは評価表に細分化されています。

各表に記録されたスコアによって、システムが Sitecore 推奨レベルを満たしている、超えている、満たしていないのどのレベルであるかが決まります。

全般的な要件/インストール済みのソフトウェア

SQL Server のバージョン	2008	2005
	+1	-1

スコア _____

SQL Server はサーバー上で実行されている唯一のアプリケーションですか？	はい	いいえ
	+1	-1

スコア _____

DMS はサーバー上に存在する唯一のデータベースですか？	はい	いいえ
	+1	0

スコア _____

ハードウェア要件

RAM	8 GB 未満	8 GB	16 GB	24 GB	32 GB 以上
スタンドアロンサーバー	-1	0	+1	+2	+3
仮想サーバー	-2	-1	0	+1	+2

スコア_____

CPU	1 コア	2 コア	4 コア	8 コア	16 コア以上
スタンドアロンサーバー	-1	0	+1	+2	+3
仮想サーバー	-2	-1	0	+1	+2

スコア_____

データ ディスクは独立していますか?	はい	いいえ
	+1	-1

スコア_____

ログ ディスクは独立していますか?	はい	いいえ
	+1	-1

スコア_____

TempDb ディスクは独立していますか?	はい	いいえ
	+1	-1

スコア_____

第 4 章

Analytics 設定パラメーター

この章では、Sitecore.Analytics.config ファイルに指定されている 2 つのパラメーターを取り上げます。どちらも情報がメモリからデータベースに書き出される頻度に影響を与えます。

2 つのパラメーターとは、Analytics.TrackerChanges.FlushInterval と Analytics.TrackerChanges.MaxRows です。

この章には次のセクションがあります。

- パラメーターの動作

4.1 パラメーターの動作

4.1.1 Analytics.TrackerChanges.FlushInterval

このパラメーターは次の動作を制御します。

- データが、データベースに書き込まれる前に、メモリに格納されている最長時間。
- (上記と同じことを別の側面から) SQL Server に対するリクエストの頻度 (低レイテンシ サーバーの問題)。
- このパラメーターの値が小さいと SQL Server に対するリクエストの頻度が高くなり、結果として、SQL Server の負荷が大きくなる。
- このパラメーターの値が大きいと、サーバーのクラッシュ/再起動時にデータが失われるリスクが高くなる(データキューがあふれておらず、フラッシュが時々発生している状態であるため)。

4.1.2 Analytics.TrackerChanges.MaxRows

このパラメーターは次の動作を制御します。

- コミット キューによる最大メモリ消費量 (間接的に制御)。
- (上記と同じことを別の側面から) サーバーのクラッシュ時に失われる最大データ量。
- このパラメーターの値が小さいとキューがすぐにあふれるため、SQL Server に対するリクエストの頻度が高くなり、結果として、SQL Server の負荷が大きくなる。
- このパラメーターの値が大きいと、メモリ消費量、およびサーバーのクラッシュ/再起動時に失われるデータ量が大きくなる。

4.1.3 FlushInterval と MaxRows の関係

FlushInterval と MaxRows の間には、ほぼ次のような関係が成り立ちます。

$$\text{MaxRows} \sim \text{RWS} * \text{FlushInterval} / 1000 \quad (1)$$

ここで、RWS (Rows per Second) は、データベースのパフォーマンスを測る尺度で、次の式で概算値を求めることができます。

- $\text{RWS} = \text{RQS} * \text{RPR}$
- ここで、RQS とは、追跡された Web サイトの毎秒のリクエスト数 (IIS/ASP.NET パフォーマンス カウンターによる計画値または実測値) です。
- RPR とは、リクエストごとの DB 行の平均書き出し数で、次の式が成り立ちます。
 - $\text{RPR} = 3 + \text{PR} + \text{PE} + \text{A}$

- ここで、現在ページ、直前ページ、現在の訪問の 3 行は常に書き出されます。
- PR は、コンテンツ ページに定義されているさまざまなプロファイルの平均数です。
- PE は、コンテンツ ページに定義されているゴールとページ イベントの平均数です。
- A は、平均的な訪問者が参加するエンゲージメント オートメーション プランの数です。

(1) が満たされると、コミット キューには、FlushInterval の期間が切れると同時に MaxRows の数だけレコードが蓄積することになり、どちらのパラメーターを設定してもシステムに同じ効果を与えることになります。

チューニング可能なパラメーターは次のとおりです。

- (SQL Server に対する負荷と SQL Server のレイテンシによる影響) vs. (データ書き込み時の遅延とメモリ消費量)

4.2 チューニング シナリオ

4.2.1 SQL Server でトラフィックを処理し、SQL Server のレイテンシを低くする

方針は、SQL Server の負荷を大きくする代わりにデータの書き込み遅延を最小化することです。たとえば、1 ~ 10 秒未満の間隔でフラッシュします。

- $\text{FlushInterval} = 1000 - 10000$
- $\text{MaxRows} < \text{RWS} * \text{FlushInterval} / 1000$

4.2.2 SQL Server でトラフィックを処理するが、SQL Server のレイテンシを高くする

方針は、より大きな間隔でフラッシュすることで SQL Server のレイテンシによる影響を最小化することです。たとえば、フラッシュの間隔を SQL Server の ping レイテンシの値よりも大きくします。

- $\text{FlushInterval} = (100-1000) * \text{SQL Server レイテンシ}$
 - たとえば、SQL Server のレイテンシが 100 ms の場合、 $\text{FlushInterval} = 10000 - 100000$
- $\text{MaxRows} > \text{RWS} * \text{FlushInterval} / 1000$
 - MaxRows の上限はメモリ消費量です。
 - たとえば、 $\text{RWS} \sim 100$
 - $\text{MaxRows} > 10000$

4.2.3 SQL Server でトラフィックを処理しない

方針は、Analytics.Sampling.Percentage の設定によって、追跡するリクエストの数を減らし、SQL Server が負荷に対応できるようにすることです。