



Sitecore CMS 6.0 - 6.4

CMS パフォーマンス チューニング ガイド

Sitecore CMS のパフォーマンス最適化のためのデベロッパー用ガイド

目次

第 1 章	イントロダクション.....	4
第 2 章	診断手順.....	5
2.1	ブラウザ ページ テスト.....	6
2.2	ページ パフォーマンスの調査 (IIS ログ使用)	17
2.3	レンダリングのパフォーマンス	22
2.4	Sitecore ログでのメモリー リークの調査.....	25
第 3 章	チューニング手順 - 全般	39
3.1	SQL Server インデックスの断片化レベルのチェック	40
3.2	SQL Server メンテナンス プランのチェック	43
3.3	データベース テーブルのクリーンアップ	48
3.4	データベース クリーンアップ エージェントのチェック.....	51
3.5	未使用の検索インデックスの無効化	55
3.6	ソフトウェアおよびサーバーの設定.....	57
第 4 章	チューニング手順 - データベースのプロパティ.....	70
4.1	SQL Server 2008 (100) に設定された互換性レベル	71
4.2	自動終了プロパティを false に設定.....	74
4.3	自動圧縮プロパティを false に設定.....	77
4.4	復旧モデルを単純に設定.....	80
第 5 章	チューニング手順 - Sitecore のキャッシュ	83
5.1	初期キャッシュ値の設定	84
5.2	Sitecore のキャッシュのチューニング	88
5.3	プリフェッチ キャッシュ設定のガイドライン	92
5.4	出力 (レンダリング) キャッシュ設定のガイドライン.....	94
第 6 章	チューニング手順 - IIS 設定	99
6.1	HTTP キープアライブの有効化	100
6.2	IIS の Expire Web コンテンツ ヘッダー	103
6.3	IIS の静的なコンテンツの圧縮の有効化	106
6.4	IIS の動的なコンテンツの圧縮の有効化 (オプション)	109
第 7 章	チューニング手順 - Sitecore のクライアント最適化	112
7.1	実行時間の長いバリエーターのチェック	113
7.2	過剰なアイテム バージョンのチェック	116
7.3	ノードごとの過剰アイテムのチェック	120
7.4	その他のクライアント固有の最適化	123

この文書に記載されている情報は、文書の公開日現在の Sitecore Corporation の見解であり、事前の予告なく変更されることがあります。この文書に記載されている内容は、現状のまま提供され、一切の保証を伴いません。また、Sitecore による提供または責務を表明するものではありません。Sitecore は、記載されている情報の正確性を保証するものではありません。Sitecore はこの文書に関し、明示または黙示を問わず、いかなる保証もしません。

この文書で他社製品に言及する記述が含まれる場合、それらはお客様の便宜上の理由により提供されています。このような記述はすべて、Sitecore による推奨またはサポートを意味するものではありません。Sitecore は、これらの記述の正確性を保証できず、また、製品はいずれ変更される可能性があります。またこれらの記述は、製品の包括的な説明を目的としておらず、製品を理解する上で重要なポイントの簡単な説明を目的としています。これらの製品の正式な記述については、製品のそれぞれのメーカーにお問い合わせください。

記載されているすべての商標は、それぞれ各社の所有物です。

©2011 Sitecore Corporation. All rights reserved.

第 1 章

イントロダクション

本ガイドは、2 つの部分からなります。最初のパートには一連の診断手順が記載されています。これは、パフォーマンスの問題を見つけるのに役立つほか、チューニングによるパフォーマンスの向上を測定する手段になります。2 番目のパートは、一連のチューニング タスクで、関係のあるセクションごとに分かれています。これは、Sitecore の実装が正常に最適のパフォーマンスで稼働していることを、確実にするのに役立ちます。

このマニュアルには次の章があります。

- **第 1 章 — イントロダクション**
- **第 2 章 — 診断手順**
- **第 3 章 — チューニング手順 - 全般**
- **第 4 章 — チューニング手順 - データベースのプロパティ**
- **第 5 章 — チューニング手順 - Sitecore のキャッシュ**
- **第 6 章 — チューニング手順 - IIS 設定**
- **第 7 章 — チューニング手順 - Sitecore のクライアント最適化**

第 2 章

診断手順

診断手順は、Sitecore の実装におけるパフォーマンスの問題を特定するのに役立つ一連のテストです。

定義された手順は、CMS チューニングを補完するものです。Sitecore CMS をチューニングする前と後に診断手順を実行することで、パフォーマンスの向上を記録することができます。

この章には次のセクションがあります。

- ブラウザー ページ テスト
- ページ パフォーマンスの調査 (IIS ログ使用)
- レンダリングのパフォーマンス
- Sitecore ログでのメモリー リークの調査

2.1 ブラウザー ページ テスト

"上位 500 の Web サイトの平均のページ サイズは、2003 年の 5 倍以上になった。平均の Web ページ サイズは 2003 年の 93.7K から、2009 年には 5.4 倍以上の 507K を越えるまでに大きくなった (図 1 参照)。(Domenech 他 2007、Flinn&Betcher 2008、Charzinsk 2010)。この 6 年の間に、平均的 Web ページ内のオブジェクトの数は、ページあたりで 25.7 から 2 倍以上の 64.7 になった。1995 年以降のより長期間の統計によると、平均的 Web ページのサイズは 35 倍に増え、ページあたりのオブジェクト数は 28 倍に増えた。"-

<http://www.websiteoptimization.com/speed/tweak/average-web-page/>

上記報告は、今日の Web ページでは、ロードされるオブジェクトの数とサイズを認識することが重要であることを意味します。また、Web サーバーの設定では、オブジェクトをキャッシュし、要求の回数を最小限にすることが重要となります。

Web ページをブラウザーにロードするときの事象を解析することによって、収集できる情報は多数あります。要求の内容、応答時間、オブジェクト サイズなどの情報を入手できます。キャッシング、圧縮、CDN の活用、キーブアライブ情報の観点から、サーバーを設定する方法を考察することもできます。

このタスクでは、オープン ソース プラグインの AOL Pagetest を利用し、併せてその結果を解釈する方法をみていきます。

2.1.1 設定

AOL Pagetest ブラウザー プラグインは、次のサイトからダウンロードできます: <http://sourceforge.net/projects/pagetest/files/>

ダウンロードしたら、デフォルトの場所にインストールします。AOL Pagetest プラグインは IE 7 以上のブラウザーで動作し、[ツール] メニューに表示されます。

メモ

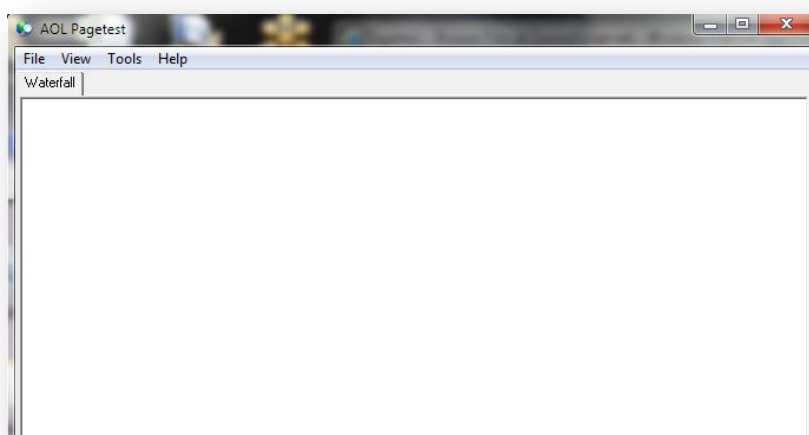
インストール中に IE が開いていた場合、AOL Pagetest を [ツール] メニューに表示するためには IE を再起動する必要があります。

2.1.2 AOL Pagetest ブラウザー プラグインの使用方法

AOL Pagetest を起動するには:

1. IE を起動します。

2. [ツール] メニューで、[AOL Pagetest] をクリックします。



2.1.3 情報の収集

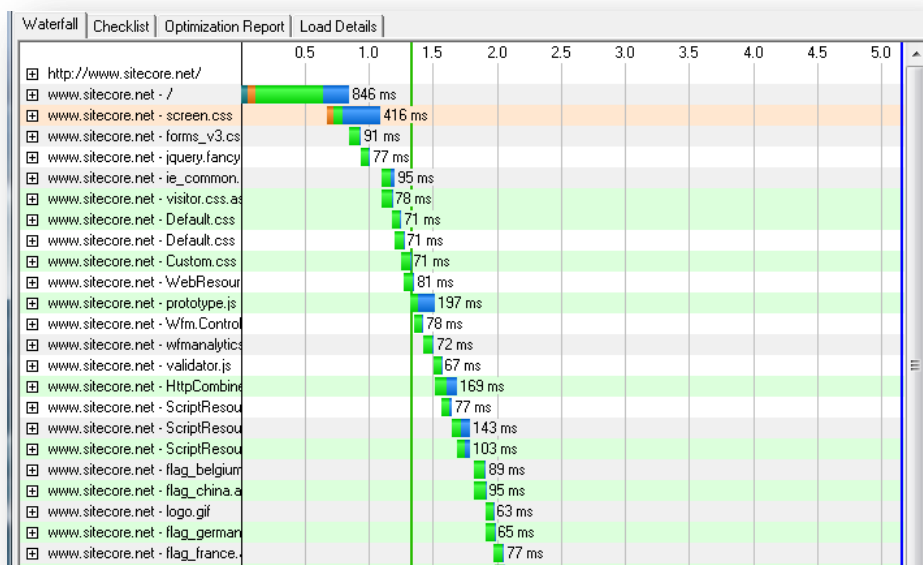
この課題では、ブラウザー キャッシュの消去、Web ページのリロード、およびキャッシュがロードされた状態での Web ページ要求で Web ページがどのように反応するかという、3 つの有用なページ ロード シナリオが用意されています。

- ブラウザー キャッシュの消去 - 使用されているブラウザーから、初めてページにアクセスした場合の動作をシミュレーションします。
 - **IE 8** 以降のバージョンで、ブラウザー キャッシュをクリアするには:
 - [セーフティ] メニューで、[閲覧の履歴の削除...] をクリックします。
 - [お気に入り Web サイトデータの保持] チェック ボックスをオフにし、[インターネット一時ファイル]、[Cookie]、および [履歴] チェック ボックスをオンにします。
 - **IE 7** で、ブラウザー キャッシュをクリアするには:
 - [ツール] メニューで、[インターネット オプション] をクリックします。
 - [閲覧の履歴] 下の [削除...] をクリックします。
 - キャッシュを削除するには、[削除] をクリックします。
 - [閉じる]、または [OK] ボタンをクリックします。
- Web ページのリロードまたは最新表示 - キャッシュ内のオブジェクトに対する要求が強制実行され、304 ステータス コードが返されます。304 ステータス コードが表示されると、ダウンロード中でない時も、オブジェクトに対する要求がなされます。
- 最近訪問された Web ページへのアクセス (URL の入力) - これにより、既にブラウザー キャッシュにロードされているアイテムは要求しなくてもアクセスされるため、一連の要求の大部分が削減されます。

1 つ目のシミュレーション — ブラウザー キャッシュの消去

1. **IE** を起動します。
2. ブラウザー キャッシュをクリアします (上記参照)。
3. AOL Pagetest を起動します。
4. テストする URL を入力します (例: `http://www.sitecore.net`)。
5. AOL Pagetest ウィンドウには、4 つのタブがあります ([**ウォーターフォール**]、[**チェック リスト**]、[**最適化レポート**]、[**ロードの詳細**])。次のセクションで、各タブに表示される情報を解釈する方法を説明します。

4 つのタブの表示:



考察点:

- [**ウォーターフォール**] タブには、Web ページ内の各オブジェクトに対する完全な要求が表示されることに注意ください。
- CSS や JS ファイルなどの資産は結合されないため、これらをダウンロードするための複数の要求がなされています。すべての CSS ファイルと JS ファイルを結合することによって、要求が 1 回で行われるため、ページをレンダリングする時間が短縮されます。

Waterfall	Checklist	Optimization Report	Load Details							
		Cache Static	Use a CDN	Combine CSS	GZIP text	mpress Imag	Keep-Alive	Cookies	Minify JS	No ETags
http://www.sitecore.net/		77%	0%	60%	100%	78%	100%	1%	76%	6%
www.sitecore.net - /					✓		✓	✓		✓
www.sitecore.net - screen.css	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - forms_v3.cs	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - jquery.fancy	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - ie_common.	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - visitor.css.a					✓		✓	⚠		✓
www.sitecore.net - Default.css	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - Default.css	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - Custom.css	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - WebResour	✓	✗	✗	✗	✓		✓	✗	✗	✓
www.sitecore.net - prototype.js	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - Wfm.Control	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - wfmanalytic	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - validator.js	⚠	✗	✗	✗	✓		✓	✗		✗
www.sitecore.net - HttpCombine	✓	✗	✗	✗	✓		✓	✗		✓
www.sitecore.net - ScriptResou	✓	✗	✗	✗	✓		✓	✗		✓
www.sitecore.net - ScriptResou	✓	✗	✗	✗	✓		✓	✗		✓
www.sitecore.net - ScriptResou	✓	✗	✗	✗	✓		✓	✗		✓
www.sitecore.net - flag_belgium	✓	✗	✗	✗	✓		✓	✗		✓
www.sitecore.net - flag_china.a	✓	✗	✗	✗	✓		✓	✗		✓
www.sitecore.net - logo.gif	⚠	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - flag_german	✓	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - flag_france.	✓	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - flag_hungar	✓	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - fancy_close	⚠	✗	✗	✗		✓	✓	✗		✗
www.sitecore.net - flag_japan.a	✓	✗	✗	✗		✓	✓	✗		✗
www.sitecore.net - flag_denmar	✓	✗	✗	✗		✓	✓	✗		✗
www.sitecore.net - fancy_nav_	⚠	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - flag_netherl	✓	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - flag_norway	✓	✗	✗	✗		⚠	✓	✗		✗
www.sitecore.net - fancy_nav_	⚠	✗	✗	✗		✓	✓	✗		✗

Waterfall | Checklist | Optimization Report | Load Details

Optimization Report: 04/11/11 at 11:11:25

Results for 'http://www.sitecore.net/':

Page load time: 5.157 seconds
 Time to first byte: 0.640 seconds
 Time to Base Page Downloaded: 0.846 seconds
 Time to Start Render: 1.333 seconds
 Time to Document Complete: 5.157 seconds
 Time to Fully Loaded: 5.157 seconds
 Bytes sent out: 62.812 KB
 Bytes received: 381.117 KB
 DNS Lookups: 1
 Connections: 2
 Requests: 106
 OK Requests: 106
 Redirects: 0
 Not Modified: 0
 Not Found: 0
 Other: 0
 Base Page Response: 200

Enable browser caching of static assets:

- WARNING (7.0 days) - http://www.sitecore.net/css/fancybox/jquery.fancybox-1.3.1.css?version=11.4.1.19924
- WARNING (7.0 days) - http://www.sitecore.net/css/forms_v3.css?version=11.4.1.19924
- WARNING (7.0 days) - http://www.sitecore.net/css/ie_common.css?version=11.4.1.19924
- WARNING (7.0 days) - http://www.sitecore.net/css/screen.css?version=11.4.1.19924
- WARNING (7.0 days) - http://www.sitecore.net/images/_interface/frontpage/sidebar_border.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_interface/frontpage/sidebar_morearrow.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_interface/icon_searchfield.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_interface/logo.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_interface/navbg.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_interface/navdivider.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_nav/downarrow.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_nav/globalnavdivider.gif
- WARNING (7.0 days) - http://www.sitecore.net/images/_sidebar/buttonbg.gif

```

Waterfall | Checklist | Optimization Report | Load Details |
Results for 'http://www.sitecore.net/':
Page load time: 5.157 seconds
Time to first byte: 0.640 seconds
Time to Base Page Downloaded: 0.846 seconds
Time to Start Render: 1.333 seconds
Time to Document Complete: 5.157 seconds
Time to Fully Loaded: 5.157 seconds
Bytes sent out: 62.812 KB
Bytes received: 381.117 KB
DNS Lookups: 1
Connections: 2
Requests: 106
OK Requests: 106
Redirects: 0
Not Modified: 0
Not Found: 0
Other: 0
Base Page Response: 200

Request details:
Request 1:
Action: GET
Url: http://www.sitecore.net/
Host: www.sitecore.net
Result code: 200
Transaction time: 0.846 seconds
Time to first byte: 0.526 seconds
Document: 1
Socket: 4
Request size (out): 570 Bytes
Response size (in): 16098 Bytes
Request Object Size (out): 0 Bytes
Response Object Size (in): 15485 Bytes
Response Object Size (bodylen): 61895 Bytes
Request Headers:
GET / HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/x-shockwave-flash
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; .NET CLR 1.1.4324.2251; .NET CLR 1.0.3705.6248; .NET CLR 1.0.3705.5512)
Accept-Encoding: gzip, deflate
Host: www.sitecore.net
Connection: Keep-Alive
Response Headers:
HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 15485
Expires: -1
Date: Mon, 11 Apr 2011 15:10:45 GMT
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Server: Microsoft-IIS/7.5
Cache-Control: no-cache, no-store
Pragma: no-cache
X-AspNet-Version: 2.0.50727
Set-Cookie: ASP.NET_SessionId=tndn12iq50x5me952kxpyej; path=/; HttpOnly
Set-Cookie: SC_ANALYTICS_GLOBAL_COOKIE=1F70BEDC8230426E85AF2DE49E2BEFCF; expires=Tue, 10-Apr-2012 15:10:45 GMT; path=/
Set-Cookie: SC_ANALYTICS_SESSION_COOKIE=33BB856FC4434B148B0086AB43E28CC4.1; path=/
X-Powered-By: ASP.NET
Vary: Accept-Encoding

```

2 つ目のシミュレーション - ブラウザー ページの更新

1. **AOL Pagetest** の **[File]** メニューで、**[New]** をクリックします。
2. ブラウザー ページを更新します — **[最新の情報に更新]** をクリックするか、F5 キーを押します。

3. [ウォーターフォール] タブで結果を確認し、1 つ目のシミュレーションの結果と比較します。



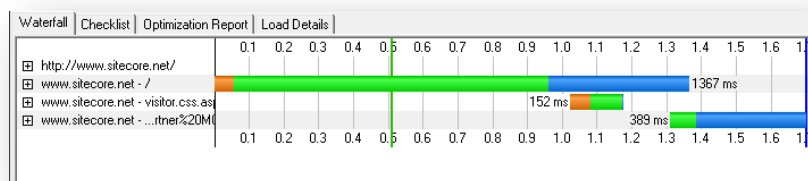
考察点:

- 黄色で強調表示されているオブジェクトは、30x ステータス コードです。
- この結果は、キャッシュされたアイテムが要求されているが、ダウンロードされていないことを示します。これは、更新されたページにとっては想定された動作です。
- ステータス コードが 304 のアイテムは、3 つ目のシミュレーション、つまり「最近訪問された Web ページ」では要求すべきでないことがこの表示からわかります。

3 つ目のシミュレーション — 最近訪問された Web ページ

1. AOL Pagetest の [File] メニューで、[New] をクリックします。
2. ブラウザーで URL を選択し、Enter キーを押します (ブラウザーの [最新の情報に更新] ボタンはクリックしません)。

3. **[ウォーターフォール]** タブで結果を確認し、1 つ目と 2 つ目のシミュレーションの結果と比較します。



考察点:

- 一連の要求は大幅に削減されています。
- 最初の 2 つの要求、「/」および「visitor.css」は、動的にロードされるもので、キャッシュはされていません。
- 3 番目のアイテムについては、キャッシュからロードされていない理由を調べる必要があります。詳細な調査の結果、これは順番に変わる画像に対する要求で、訪問ごとに変わるものであることがわかりました。(キャッシュにロードされていなくても構わないものです)。









2.1.4 結果の理解

ここでは、**[ウォーターフォール]**、**[チェック リスト]**、**[最適化レポート]**、**[ロードの詳細]** タブについて詳しく見ていき、表示される情報を説明します。

[ウォーターフォール] タブ

[ウォーターフォール] タブでは、要求された Web ページのオブジェクトを構成する一連の要求が表示されます。要求オブジェクトは左下に一覧表示され、時間はその上に一覧表示されます。

要求/応答を分類表示する色は、次のとおりです:

Key	
	DNS Lookup
	Initial Connection
	Time to First Byte
	Content Download
	Start Render
	Document Complete
	3xx Result
	4xx Result

- **[DNS ルックアップ]**— 要求された URL の IP アドレスを検索するのにかかる時間です。
- **[初期接続]**— ソケットを開くために、クライアントからサーバーへ接続する時間です。要求ごとに初期接続される場合、HTTP キープアライブがサーバー上で有効にされていない可能性があることにご注意ください。
- **[最初のバイトまでの時間]**— 要求から、オブジェクトの最初のバイトを受信するまでの時間です。[最初のバイトまでの時間] が長い場合は、サーバーに問題があるか（すなわち、パフォーマンスの問題）、ネットワークに問題がある可能性があります。
- **[コンテンツのダウンロード]**— オブジェクトをダウンロードするのに必要な時間です。[コンテンツのダウンロード] が長い場合は、オブジェクトが大きいことを意味する可能性があります。オブジェクト サイズの確認が必要である可能性があります。次はその例です:最適化や圧縮のされていない画像は、Web ページのパフォーマンスに影響を及ぼす可能性があります。
- **[レンダリングの開始]**— [ウォーターフォール] タブに表示される緑の縦線です。ブラウザーに表示されたコンテンツを、ユーザーが閲覧し始めたことを意味します。技術的には、Web ページの高さと幅が 0 より大きくなった時刻を示します。
- **[ドキュメント終了]**— ドキュメント終了イベントの開始時間です。
- **[3xx 結果]**— 黄色に強調表示されたオブジェクトは、要求やキャッシュ要求などの 3xx ステータス コードが返されたことを意味します。
- **[4xx 結果]**— 赤色に強調表示されたオブジェクトは、4xx ステータス コード、またはエラー コードが返されたこと意味します。これらは、早急に調査する必要があります。

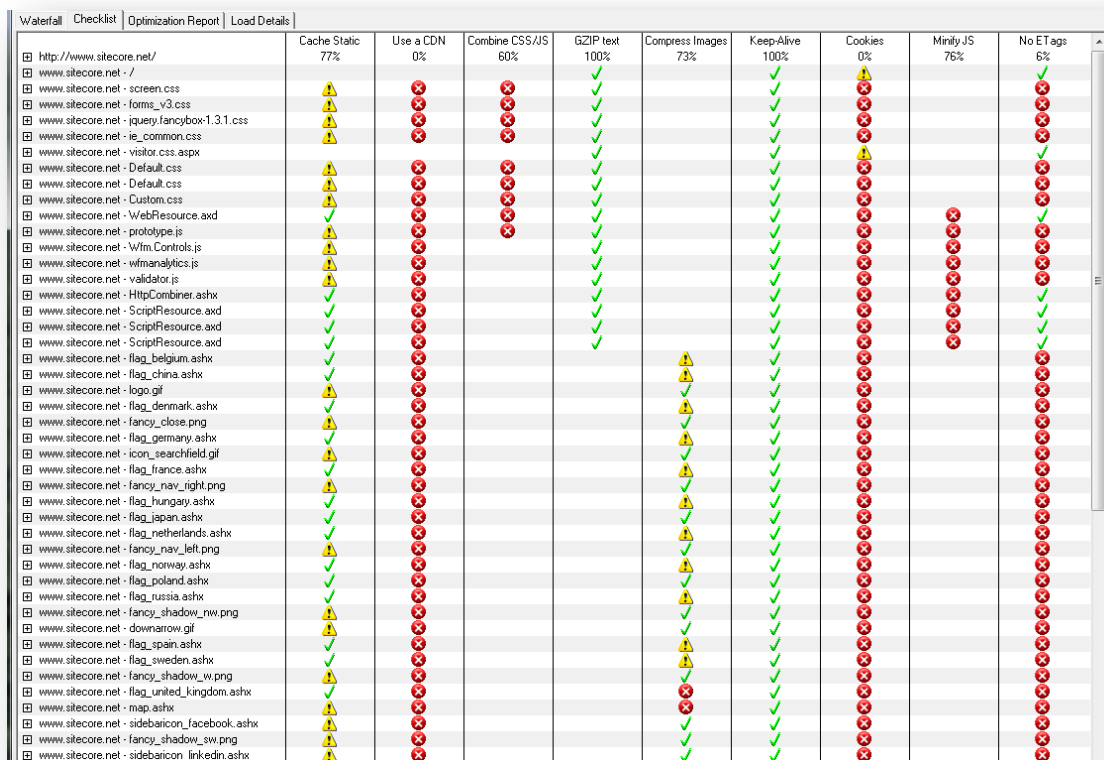
[チェック リスト] タブ

[チェック リスト] タブは、設定および手順に関連するパフォーマンスを、サイトがいかに活用しているかを示すレポートカードです。

題目は、重要な順に左から右に一覧表示されます:

- 緑色のチェック マークは、成功を意味します。
- 黄色の三角は、警告を意味します。

- 赤色の X は、"不成功" を意味します。



次の表には、各トピックの意味、適用先のオブジェクト、および確認事項についての情報が含まれます：

[静的なキャッシュ]	適用可能オブジェクト	MIME タイプが「text/*」、「*javascript*」、または「mage/*」の、html 以外のあらゆるオブジェクトで、明示的に 0 か -1 の Expires ヘッダー、「private」、「no-store」、もしくは「no-cache」のキャッシュ制御ヘッダー、または「no-cache」の Pragma ヘッダーもたないもの。
	確認事項	"Expires" ヘッダーが表示されているか (かつ、0 や -1 ではない)、または "cache-control: max-age" ディレクティブが表示されており、1 時間以上に設定されています。有効期限が 30 日間未満に設定されている場合、警告が返されます (現時点では、max-age に対してのみ適用されます)。
[CDN の使用]	適用可能オブジェクト	html 以外のすべての静的コンテンツ (css、js、および画像)
	確認事項	既知の CDN (既知の CDN ネットワークにマップされている CNAM E) 上にホストされているかどうかを確認します。既知の CDN には、Akamai、Amazon CloudFront、Coral Cache、Edgecast、Google、Highwinds、Internap、Limelight、Mirror Image、Panther and、および Yahoo があります。

[CSS/JS の結合]	適用可能オブジェクト	すべての css と JavaScript オブジェクト
	確認事項	同じタイプのファイルが複数ある場合、2 番目以降の各 css ファイルでは 5% 削減され、2 番目以降の各 JavaScript ファイルでは 10% 削減されます。
[GZIP テキスト]	適用可能オブジェクト	MIME タイプが「text/*」か「*javascript*」のすべてのオブジェクト
	確認事項	gzip の場合、Transfer-Encoding がチェックされているかどうかを確認します。チェックされていない場合、ファイルは圧縮され、圧縮率が結果となります（そのため、圧縮によってテキスト サイズの 30% を削減できるページでは、70% のテスト結果を得られます）
[画像の圧縮]	適用可能オブジェクト	MIME タイプが「image/*」のすべてのオブジェクト
	確認事項	GIF - すべて成功 PNG - 8 ビット以下である必要があります（24-ビットの PNG は成功しません）。 JPEG - フォトショップ品質 50 では、10% 以下は成功し、10% ~ 50% は警告、それより大きければ失敗します。 総合スコアは、画像の再圧縮によって削減できる画像のバイト数の割合です。
[キープアライブ]	適用可能オブジェクト	ページに複数のオブジェクトを提供するドメインからの、すべてのオブジェクト — たとえば、単一のオブジェクトのみが規定のドメインから提供される場合はチェックされません。
	確認事項	応答ヘッダーが「キープアライブ」ディレクティブを含んでいるか、所定のホストからの複数オブジェクトに同一ソケットが使用されました。
[Cookie]	適用可能オブジェクト	すべての要求
	確認事項	Cookie を送信する静的オブジェクトへの任意の要求が失敗するかどうかを確認します。Cookie を送信する他のすべての要求では警告が返されます。
[JS の圧縮]	適用可能オブジェクト	すべての html、JavaScript、および json 応答
	確認事項	JSMIn を用いて、JavaScript を実行します。元のコンテンツが gzip でエンコードされた場合、圧縮版も比較のために gzip 圧縮されます。5KB より多くか 10% より多くが削減される場合、失敗します。1KB よりも多くが削減される場合、警告になります。それ以外は成功します。
[Etag がない]	適用可能オブジェクト	すべての要求
	確認事項	応答ヘッダーが ETag ヘッダー を含む場合、要求は失敗します。

不成功と警告を解決するための情報:

- [静的なキャッシュ] 列内の不成功は、HTTP Expires Web コンテンツ ヘッダーが設定されていないことを意味します。「IIS Expire Web コンテンツ ヘッダー」を参照してください。

- [静的なキャッシュ] 列内の警告は、HTTP Expires Web コンテンツ ヘッダーの値が 30 日間未満に設定されていることを意味します。次の項を参照してください。IIS の Expire Web コンテンツ ヘッダー
- 結合 CSS/JS 列内の不成功は、CSS および/または JS ファイルが結合されてファイル数が少なくなっていないことを意味します。これらのオブジェクトを引き込むように複数の要求がなされることを意味します。
- [GZIP テキスト] 列内の不成功は、静的圧縮が Web サーバー上で有効でないことを意味します。「IIS 静的コンテンツの圧縮の有効化」を参照してください。
- [キーブアライブ] 列内の不成功は、HTTP キーブアライブが Web サーバー上で有効でないことを意味します。このため、要求されたオブジェクトごとに接続が開始される必要が生じます。次の項を参照してください。HTTP キーブアライブの有効化

[最適化レポート] タブ

[最適化レポート] タブで、ページのロード、および改良できる部分 ([チェックリスト] タブ内のトピックに基づく) に関する全般情報を提供します。場合によっては、圧縮の設定、すなわち削減できるおよそのサイズを提供します。

[最適化レポート] タブは、[ウォーターフォール] および [チェック リスト] の両方のタブに表示されるテキスト表示です。

[ロードの詳細] タブ

[ロードの詳細] タブは、要求ごとに生じることの詳細な説明です。この情報は、特定の要求に関する課題に絞り込むのに有用です。

2.2 ページ パフォーマンスの調査 (IIS ログ使用)

ページ パフォーマンスを調査するタスクは、往復で 4000 ミリ秒 (4 秒) 以上かかるページを、IIS ログを用いて検索するように設計されています。結果は記録され、詳細な調査の出発点として用いられます。詳細な調査では、レンダリング コンポーネント (1 つあるいは複数) が、低いパフォーマンスを生じる原因になる可能性が検討されます。

ページのパフォーマンスを直接測定する場合に対して、IIS ログの解析情報を使用することの利点は、位置情報 (位置によって起こりうるネットワークの問題)、および使用ピーク時に起こりうる問題 (容量の問題の目安) も参照できることです。

2.2.1 必要なスキル

- Log Parser Lizard GUI の実務知識
- IIS ログの実務知識

2.2.2 症状

- 往復するのに時間がかかるページ。

2.2.3 パートナー/お客様との対話

パートナー/お客様/Web サイトの所有者が、表示が遅く見えるあらゆるページに気づいている場合は、彼らに尋ねてみるのが、問題/ロードに時間のかかるページがどこにあるかを見つけ出す最初のステップです。パフォーマンスの問題の原因についての詳細な調査のためには、URI を記録します。

Web サイトの使用ピーク時間も調べます。使用ピーク時にロード処理をするための容量が Web サイトにあるかどうかを調べるのに、この情報は用いられます。

2.2.4 時間のかかる要求の IIS ログへのクエリの手順

要求を降順で処理するのに必要な時間によって順序付けられる、要求された URL を、このクエリで調べます。調査対象のレンダリングやレンダリングの設定に的を合わせられます。

Sitecore クライアント ツールのトラフィックを調べないように、/sitecore/ を含むあらゆる URL は除外することに、ご注意ください。

IIS ログをクエリするために、時間のかかる要求を検索します：

1. **Log Parser Lizard GUI** を起動します。
2. **新規クエリ**を作成します。
3. デフォルトのログのフォーマットは **IIS ログ**入力フォーマットに設定されます。

Run the following script(change the **#logs location#**):

```
SELECT TO_TIMESTAMP(date, time) as Timestamp, cs-uri-stem as [URI, c-ip AS [Client IP], time-taken as [Time],s
c-status as [Status]
FROM #logs location#
WHERE ((EXTRACT_EXTENSION(URI) = 'aspx' OR EXTRACT_EXTENSION(URI) = '' ) AND URI NOT LIKE '%/sitecor
e%')
AND time-taken >4000
AND Status = '200'
ORDER BY time-taken DESC
```


2.2.5 結果の理解

結果は 2 つのパートに表示されます。パート 1 は、すべては同一のクライアント IP アドレスからくる、一連の時間のかかる要求についてです。これは、組織内のテストを示す可能性があります。パート 2 は、外部トラフィックからサイトにくる、より典型的な結果です。

結果 パート 1

Query					
Timestamp	URI	Client IP	Time	Status	
27.01.2011 16:24:40	/Sverige.aspx	88.131.15.18	567,640	200	
30.01.2011 08:35:30	/en/Norge.aspx	195.184.101.130	468,644	200	
30.01.2011 08:35:30	/Danmark.aspx	195.184.101.130	468,363	200	
30.01.2011 08:35:31	/Sverige.aspx	195.184.101.130	467,082	200	
30.01.2011 08:35:31	/Sverige.aspx	195.184.101.130	466,504	200	
30.01.2011 08:35:32	/	195.184.101.130	451,616	200	
30.01.2011 08:35:33	/en/Norge.aspx	195.184.101.130	445,305	200	
30.01.2011 08:35:32	/Danmark.aspx	195.184.101.130	444,945	200	
30.01.2011 08:35:34	/Sverige.aspx	195.184.101.130	443,945	200	
30.01.2011 08:35:33	/Sverige.aspx	195.184.101.130	443,492	200	
30.01.2011 08:35:36	/UnitedKingdom.aspx	195.184.101.130	437,650	200	
30.01.2011 08:35:36	/	195.184.101.130	437,462	200	
30.01.2011 08:35:35	/Danmark.aspx	195.184.101.130	436,869	200	
30.01.2011 08:35:35	/UnitedKingdom.aspx	195.184.101.130	436,791	200	
30.01.2011 08:35:34	/en/Norge.aspx	195.184.101.130	436,588	200	
30.01.2011 08:35:36	/	195.184.101.130	422,122	200	

クライアントの IP アドレスの位置を調べて確認するために、<http://ip2location.com/1.2.3.4> にアクセスします (「1.2.3.4」はクライアントの IP アドレスに置き換えます)。たとえば、<http://ip2location.com/195.184.101.130> となります (注: これは、位置に基づいて起こりうるネットワークの問題を見つけ出すのにも有用です):

IP Address	Country	Region	City	Latitude/ Longitude	ZIP Code	Time Zone	
195.184.101.130	 DENMARK	KOBENHAVN	COPENHAGEN	55.676294 12.568116	-	+01:00	
	Net Speed	ISP		Domain			
	COMP	RESULTMAKER A/S		RESULTMAKER.COM			
	IDD Code	Area Code	Weather Station				Map It
	45	-	DAXX0009 - COPENHAGEN				
	MCC	MNC	Mobile Brand				
	-	-	-				

結果 パート 2

4 second threshold - IIS Logs					
Timestamp	URI	Client IP	Time	Status	
31.01.2011 10:13:48	/en/Company/Contact/Japan.aspx	173.203.158.156	5,170	200	
31.01.2011 11:59:07	/News/RSS/Feeds/Denmark-News.aspx	193.3.234.5	5,155	200	
30.01.2011 21:54:23	/Danmark.aspx	195.184.101.130	5,139	200	
30.01.2011 11:54:36	/de/Hungary.aspx	78.46.71.246	5,121	200	
27.01.2011 10:36:27	/en/Products/Resources/Tours.aspx	65.61.164.180	5,092	200	
31.01.2011 10:17:45	/en/Customers/Selected-Customers.aspx	173.203.158.156	5,077	200	
30.01.2011 11:57:48	/en/Solutions/Best-CMS-Solutions-Education.aspx	78.46.71.246	5,076	200	
28.01.2011 01:09:32	/en/Japan.aspx	166.205.138.71	5,061	200	
31.01.2011 10:14:02	/en/Products.aspx	173.203.158.156	5,046	200	
29.01.2011 10:47:10	/en/Partners.aspx	65.61.143.45	5,030	200	
27.01.2011 07:33:24	/products/resources/whitepapers/gartner-magic-quadrant.aspx	202.155.14.116	5,030	200	
31.01.2011 10:14:10	/en/Partners/Hosting-Partners.aspx	173.203.158.156	4,999	200	
31.01.2011 10:17:50	/en/Customers/Selected-Customers.aspx	173.203.158.156	4,983	200	
28.01.2011 10:37:16	/en/Products/Industry-Commentary.aspx	64.39.4.224	4,983	200	
29.01.2011 10:50:27	/en/Customers.aspx	65.61.143.45	4,983	200	
31.01.2011 10:16:31	/Japan.aspx	173.203.158.156	4,983	200	
30.01.2011 11:58:36	/en/News/NewsAndEvents.aspx	78.46.71.246	4,982	200	
30.01.2011 11:56:21	/en/Company/Contact.aspx	78.46.71.246	4,966	200	
28.01.2011 15:37:03	/Sverige/Partners/KnowIT.aspx	67.195.37.153	4,952	200	
27.01.2011 10:35:45	/en/Partners/Become-Partner.aspx	65.61.164.180	4,952	200	

テストの実行、または位置のせいで起こりうるネットワークの問題のいずれかに由来する URI を特定した後で、残りを記録します。

使用ピーク時に、しきい値 (この例では 4 秒) を超える URI の数が多い場合、Web サイトにロードを処理するための十分な容量がないことを示す可能性があります。

2.2.6 推奨事項

Sitecore は、任意の .aspx ページの往復時間を 4 秒未満にすることを推奨しています。

調査結果レポート

結果の記録

Web サイトの使用ピーク時: _____

URI で往復時間が 4 秒を超えるものはすべて記録します。これらの URL は、レンダリング コンポーネントの分析タスクの実行時に使用されます。

メモ

Log Parser Lizard GUI の結果は、クリーンアップとこの文書への貼り付け用に、Excel にエクスポートすることができます。

タイムスタンプ	URI	クライアント IP	ステータス コード	時間

しきい値の 4 秒を超えた要求があります __ はい __ いいえ

ピーク使用時に異なる URL が多数あります __ はい __ いいえ

しきい値の 4 秒を超えた要求があります = いいえ

OK。往復時間が 4 秒を超えた要求がありません。

しきい値の 4 秒を超えた要求があります = はい

エラー。往復時間が 4 秒を超えた要求があります。詳細な調査が必要です。レンダリング コンポーネントのパフォーマンス タスクを参照してください。

使用ピーク時に異なる URL が多数あります = はい

エラー。使用ピーク時に、しきい値の 4 秒を超えた異なる URI が多数あります。これは、容量の問題を示す可能性が

あり、詳細な調査を必要とします。

2.3 レンダリングのパフォーマンス

Sitecore の統計ページを使用することで、レンダリングに関する情報を収集することができます。

統計ページには、ページ内かサイト全体（どの時点で統計ページを参照するかによる）で使用される各種レンダリングについての、次の情報が表示されます。

- レンダリング — レンダリングの名前。
- サイト — レンダリング用の情報を収集中のサイトの名前。
- カウント — 統計ページが最後にリセットされてから、レンダリングが呼ばれた回数。
- キャッシュからの読み込み — レンダリングがキャッシュから読み込まれる回数。
- 平均時間（ミリ秒） — レンダリングから出力までにかかる平均時間。
- アイテムの平均数 — レンダリングに含まれるアイテムの平均数。
- 最大時間 — レンダリングから出力までにかかる最大時間。
- アイテムの最大数 — レンダリングに含まれるアイテムの最大数。
- 合計時間 — 統計ページが最後にリセットされてから、このレンダリングのすべてのインスタンスにかかる時間の合計。
- アイテムの合計数 — 統計ページが最後にリセットされてから、このレンダリングのすべてのインスタンスに含まれるアイテムの合計数。
- 最後の実行 — 統計が収集された最後の時間。

2.3.1 必要なスキル

- Sitecore 統計.aspx ページの実務知識。

2.3.2 症状

- 往復するのに時間がかかる Web ページ。

2.3.3 Sitecore 統計ページを使用する手順

この手順では、/sitecore/admin フォルダー内の aspx ページにアクセスする権限を必要とします。また、統計ページは別のタブかブラウザのウィンドウから開くことをお勧めします。これにより、調査中のサイトを移動するために他のタブかウィンドウを使用している間に、統計ページを開いておくことができます。

統計ページを起動するには：

1. Web ブラウザー ウィンドウかタブのどちらかを 2 つずつ起動します。これらを、**[統計]** ウィンドウ、**[サイト]** ウィンドウと呼びます。
2. **[統計]** ウィンドウでは、`http://<site>/sitecore/admin/stats.aspx` にアクセスします。

3. [統計] ウィンドウでは、統計ページが最後にリセットされてから、要求されたレンダリングについての情報が示されます。最初からやり直したい場合は、[更新] ボタンをクリックします (たとえば、個々の Web ページに関する情報を表示したい場合、統計ページを空にする必要があります)。
4. 最後にリセットされてから収集されたレンダリングの統計を表示する場合は、提示された情報を使用します。
5. 単独のページについての情報を収集したい場合は、統計ページをリセットします。[サイト] ウィンドウでは、この単独ページが含むレンダリングについての統計を収集しようとしている Web サイトか Web ページにアクセスします。メモ: キャッシングと平均値について情報を収集できるように、ページを複数回呼び出します。
6. 統計ページの表中の情報を Excel にエクスポートすることを、お勧めします。これにより、データ分析をよりたやすくするために、情報をソートすることができます。エクスポートするには、表を右クリックし、[Microsoft Excel にエクスポート] をクリックします。

2.3.4 結果の理解

次の統計の表が Excel にエクスポートされ、最大時間の順に並び替えられ、いくつかの列は表示されません。

	A	B	C	D	E	G
1	Rendering	Site	Count	From cache	Avg. time (ms)	Max. time
2	Placeholder: content	nicam	20	0	185.3304	1670.1252
3	Sublayout: /layouts/Nicam/HomePageContent.ascx	nicam	2	0	1080.6819	1669.1879
4	Placeholder: rightcolumn	nicam	15	0	142.9569	1437.7889
5	/xsl/Nicam/NewsSpot.xslt	nicam	2	0	938.391	1419.1291
6	Sublayout: /layouts/Nicam/ThreeColumnContent.ascx	nicam	13	0	86.9166	362.6216
7	Placeholder: centercolumn	nicam	18	0	50.8885	328.2586
8	Sublayout: /layouts/Nicam/ProductForums.ascx	nicam	1	0	327.9864	327.9864
9	Sublayout: /layouts/Nicam/TwoColumnContent.ascx	nicam	5	0	82.8207	274.0745
10	Sublayout: /layouts/Nicam/ContactUsFormWrapper.ascx	nicam	1	0	260.099	260.099
11	FormRender1 (FormRender)	nicam	1	0	260.0562	260.0562
12	Placeholder: forum-content	nicam	1	0	178.4444	178.4444
13	Sublayout: /sitecore modules/Web/YAF/YAF_Forum.ascx	nicam	1	0	178.4045	178.4045
14	Placeholder: phxml	nicam	2	0	86.4851	162.7259
15	/xsl/FlashImageIterator/Flash_XMLOutput.xslt	nicam	2	0	86.4505	162.6944
16	/xsl/Nicam/Logo.xslt	nicam	21	7	12.5506	146.9511
17	Placeholder: frontpagebottomspotbar	nicam	2	0	84.1091	142.2216
18	Sublayout: /layouts/Nicam/Spots Three Column.ascx	nicam	4	0	42.0658	142.2035
19	Placeholder: spotbarcenter	nicam	4	0	33.869	121.6814
20	/xsl/Nicam/RotateSpots.xslt	nicam	6	0	32.8522	121.6473
21	/xsl/Nicam/FlexiblePersonalizationSpot.xslt	nicam	9	0	13.5927	111.2867
22	/xsl/Nicam/Top Menu.xslt	nicam	21	0	17.2713	106.0543
23	/xsl/Nicam/Product Catalog.xslt	nicam	12	0	17.6432	100.0303
24	Placeholder: frontflash	nicam	2	0	45.6854	89.0685

この表から、次の考察点が得られます:

- [最大時間] が 100 ミリ秒を上回るすべてのレンダリングについて、推奨されるコーディングがなされているかどうかを確認する必要があります。
- [キャッシュからの読み込み] 列の 0 は、キャッシュのレンダリング (HTML 出力) が設定されていないことを意味します。

2.3.5 推奨事項

レンダリングの [最大時間] を 100 ミリ秒未満にすること、およびキャッシュのレンダリング (HTML 出力) を有効に設定することをお勧めします。

調査結果レポート

結果の記録

レンダリングの [最大時間] は 100 ミリ秒未満です: __ はい __ いいえ

[キャッシュからの読み込み] 値が 0 のものが複数あります: __ はい __ いいえ

レンダリングの [最大時間] は 100 ミリ秒未満です = はい:

OK。[最大時間] が 100 ミリ秒を超えるレンダリングはありません (推奨プラクティスによる)。

レンダリングの [最大時間] は 100 ミリ秒未満です = いいえ:

エラー。[最大時間] が 100 ミリ秒を超えるレンダリングがあります。推奨プラクティスは、レンダリングの [最大時間] を 100 ミリ秒未満に保つことです。

[キャッシュからの読み込み] 値が 0 のものが複数あります = いいえ:

OK。推奨プラクティスに従って、キャッシュのレンダリング (HTML 出力) が有効で、設定されています。

[キャッシュからの読み込み] 値が 0 のものが複数あります = はい:

エラー。キャッシュのレンダリング (HTML 出力) が有効でないかあるいは設定されていません (あるいはその両方)。推奨プラクティスは、サイトのパフォーマンスを上げるために、キャッシュのレンダリング (HTML 出力) を有効に設定することです。レンダリング キャッシュを有効に設定する方法の詳細については、『キャッシュ構成リファレンス』および『プレゼンテーション コンポーネント リファレンス』マニュアルを参照してください。

2.4 Sitecore ログでのメモリー リークの調査

Sitecore システムは、Sitecore ログ ファイルに 10 分間隔で記録される一連のパフォーマンス カウンターを提供します:

- Process¥Private Bytes
- Process¥Virtual Bytes
- Process¥Page File Bytes
- .net CLR Memory¥# Bytes in all Heaps
- .net CLR Memory¥% Time in GC
- .net CLR Memory¥Large Object Heap size
- .net CLR Loading¥Bytes in Loader Heap
- .net CLR Loading¥Current Assemblies

Sitecore パフォーマンス カウンターが使用できない場合は、次のサイトからダウンロード可能です:

<http://sdn.sitecore.net/upload/sdn5/faq/administration/sitecorecounters.zip>

次のサイトの記事では、パフォーマンス カウンターに関連する問題およびその解決について説明しています: <http://sdn.sitecore.net/Scrapbook/Working%20with%20Sitecore%20Performance%20Counters.aspx>

このタスクで興味を引く 2 つのカウンターは、*Process¥Private Bytes* および *.net CLR Memory¥# Bytes in all Heaps* です。

Process¥Private Bytes カウンターは、プロセス (w3wp.exe) に排他的に割り当てられるすべてのメモリーをレポートします。システム上で、他のプロセスと共有できません。*.net CLR Memory¥# Bytes in all Heaps* カウンターは、Gen0、Gen1、Gen2、および大きなオブジェクト ヒープの結合された合計サイズをレポートします。

通常、Private Bytes および # Bytes in all Heaps は、同じ割合で増減します。Private Bytes が増加しているのに、# Bytes in all Heaps に変化がない場合、アンマネージ メモリーがリークします。両方とも増加していて、クリアされない場合、マネージ メモリーからのリークの可能性があります。

Sitecore ログ ファイル (*Log Parser Lizard GUI* と *Excel*) を使用することで、これら 2 つのカウンターをグラフにして、リークの可能性を探します。

2.4.1 必要なスキル

- Sitecore ログの実務知識
- SQL スクリプトの実務知識
- Log Parser Lizard GUI の実務知識

- Microsoft Excel のグラフの実務知識

2.4.2 症状

- OutOfMemory 例外
- IIS アプリケーション プールのリサイクル
- メモリー使用量増加につれ、低くなるパフォーマンス

2.4.3 Log Parser Lizard GUI を使用した Sitecore ログの構文解析

Log Parser ソフトウェアのインストールと設定についての情報は、次のサイトに載っています: <http://www.theclie ntview.net/?p=51>

このタスクでは、2 つの異なる結果へのクエリを実行する必要があります。最初のクエリは、Process¥Private Bytes メッセージ、および解析した値、すべてに対して実行されます。2 番目のクエリは、.net CLR Memory¥# Bytes in all Heaps、および解析した値、すべてに対して実行されます。

結果は Excel にコピーされ、グラフにして比較することが可能になります。

すべての Process¥Private Bytes へのクエリ

このクエリは、Health.Counter および Process¥Private Bytes を含むメッセージ フィールドに関するすべてのログ 入力を解析します。ログ ファイル名、ログ入力時間、パフォーマンス カウンターの名前、および値 (バイト数) に関する情報が表示されます。

DateTime	Counter	Bytes
22.12.2010 09:48:21	Private Bytes	776,396,800
22.12.2010 09:58:26	Private Bytes	825,212,928
22.12.2010 10:08:31	Private Bytes	848,396,288
22.12.2010 10:18:36	Private Bytes	963,715,072
22.12.2010 10:28:41	Private Bytes	979,030,016
22.12.2010 10:38:46	Private Bytes	928,858,112
22.12.2010 10:48:50	Private Bytes	963,055,616
22.12.2010 10:58:55	Private Bytes	1,073,197,056
22.12.2010 11:09:00	Private Bytes	1,151,533,056
22.12.2010 11:19:05	Private Bytes	1,341,677,568
22.12.2010 11:29:10	Private Bytes	1,182,515,200

1. **Log Parser Lizard GUI** を起動します。
2. **新規クエリ**を作成します。
3. ログ入力フォーマットを **Multiline Regex (Log4Net)** 入力フォーマットに変更します。
4. 次のスクリプト (**#logs location#, for example "c:¥logs¥*.txt"**) を実行します:

```
SELECT DateTime, Counter, TO_INT(Value) AS Bytes
USING
/* Parse the date from the Filename field */
INDEX_OF(Filename, 'log') AS startDate,
SUBSTR(Filename, ADD(startDate,4), 8) AS tempDate,

/* convert date to MM/DD/YYYY format */
SUBSTR(tempDate, 0, 4) AS Year,
SUBSTR(tempDate, 4, 2) AS Month,
SUBSTR(tempDate, 6, 2) AS Day,
STRCAT(Month, '/') AS datePart1,
STRCAT(datePart1, Day) AS datePart2,
STRCAT(datePart2, '/') AS datePart3,
STRCAT(datePart3, Year) AS strDate,

/* create date time string */
STRCAT(strDate, ' ') AS strDateTimePart1,
STRCAT(strDateTimePart1, Time) AS strDateTime,

/* Create TimeDate Stamp */
TO_TIMESTAMP(strDateTime, 'MM/dd/yyyyHH:mm:ss') AS DateTime,

/* Parse the name of the counter from the Message field */
LAST_INDEX_OF(Message, '/') AS endCounter,
INDEX_OF(Message, '¥¥') AS startCounter,
SUBSTR(Message, ADD(startCounter,1), SUB(SUB(endCounter,startCounter),1)) AS Counter,

/* Parse the number of bytes from the Message field */
LAST_INDEX_OF(Message, ' ') as startRawValue,
EXTRACT_TOKEN(SUBSTR(Message, startRawValue), 1, ' ') as rawValue,
REPLACE_CHR(rawValue, ',', '') AS Value

FROM #logs location#
WHERE Message LIKE '%health.counter%' AND Message LIKE '%private bytes%'
```

5. [ツール] メニューで、[グリッドを Excel にエクスポート] をクリックします。意味のある名前を付けて保存し、この xls ファイルを開きます。

すべての # Bytes in all Heaps へのクエリ

このクエリは、Health.Counter および .net CLR Memory¥# Bytes in all Heaps を含むメッセージ フィールドに関するすべてのログ入力を解析します。ログ ファイル名、ログ入力時間、パフォーマンス カウンターの名前、および値 (バイト数) に関する情報が表示されます。

DateTime	Counter	Bytes
22.12.2010 09:48:21	# Bytes in all Heaps	471,164,984
22.12.2010 09:58:26	# Bytes in all Heaps	588,572,200
22.12.2010 10:08:31	# Bytes in all Heaps	646,315,600
22.12.2010 10:18:36	# Bytes in all Heaps	744,769,840
22.12.2010 10:28:41	# Bytes in all Heaps	673,051,952
22.12.2010 10:38:46	# Bytes in all Heaps	710,841,824
22.12.2010 10:48:50	# Bytes in all Heaps	743,166,456
22.12.2010 10:58:55	# Bytes in all Heaps	843,086,048
22.12.2010 11:09:00	# Bytes in all Heaps	915,655,080
22.12.2010 11:19:05	# Bytes in all Heaps	1,070,248,360
22.12.2010 11:29:10	# Bytes in all Heaps	924,331,240
22.12.2010 11:39:15	# Bytes in all Heaps	962,719,544

1. **Log Parser Lizard GUI** を起動します。
2. **新規クエリ**を作成します。
3. ログ入力フォーマットを **Multiline Regex (Log4Net)** 入力フォーマットに変更します。
4. 次のスクリプト (`#logs location##` を、たとえば `"c:¥ログ ストリーム"` に変更) を実行します:

```

SELECT DateTime, Counter, TO_INT(Value) AS Bytes
USING
/* Parse the date from the Filename field */
INDEX_OF(Filename, 'log') AS startDate,
SUBSTR(Filename, ADD(startDate,4), 8) AS tempDate,

/* convert date to MM/DD/YYYY format */
SUBSTR(tempDate, 0, 4) AS Year,
SUBSTR(tempDate, 4, 2) AS Month,
SUBSTR(tempDate, 6, 2) AS Day,
STRCAT(Month, '/') AS datePart1,
STRCAT(datePart1, Day) AS datePart2,
STRCAT(datePart2, '/') AS datePart3,
STRCAT(datePart3, Year) AS strDate,

/* create date time string */
STRCAT(strDate, ' ') AS strDateTimePart1,
STRCAT(strDateTimePart1, Time) AS strDateTime,

/* Create TimeDate Stamp */
TO_TIMESTAMP(strDateTime, 'MM/dd/yyyyHH:mm:ss') AS DateTime,

/* Parse the name of the counter from the Message field */
LAST_INDEX_OF(Message, ')') AS endCounter,
INDEX_OF(Message, '¥¥') AS startCounter,
SUBSTR(Message, ADD(startCounter,1), SUB(SUB(endCounter,startCounter),1)) AS Counter,

```

```

/* Parse the number of bytes from the Message field */
LAST_INDEX_OF(Message, ' ') as startRawValue,
EXTRACT_TOKEN(SUBSTR(Message, startRawValue), 1, ' ') as rawValue,
REPLACE_CHR(rawValue, ',', '') AS Value

```

FROM #logs location#

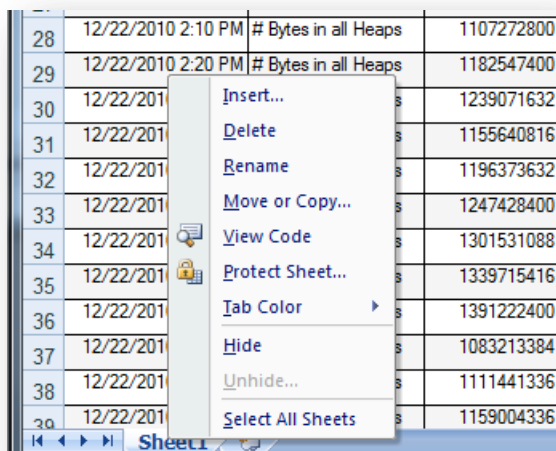
WHERE Message LIKE '%health.counter%' AND Message LIKE '%bytes in all heaps%'

5. [ツール] メニューで、[グリッドを Excel にエクスポート] をクリックします。temp.xls と名前を付けて保存し、この xls ファイルを開きます。

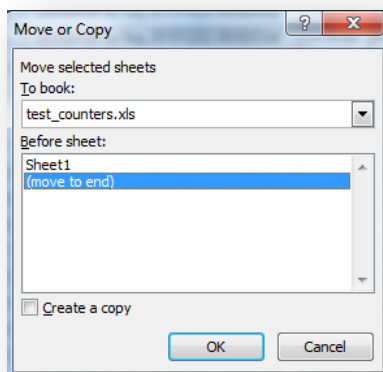
2.4.4 結果のグラフ化

Excel でグラフを作成する方法は、いくつかあります。以下は、Process¥Private Bytes の結果と .net CLR Memory¥# Bytes in all Heaps を比較するための折れ線グラフを作成するステップです。

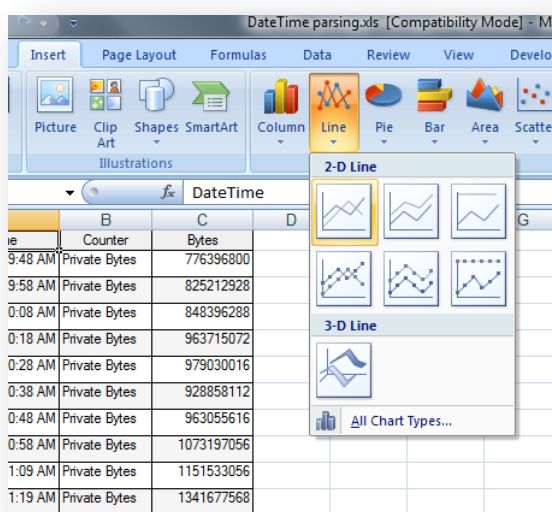
1. 最初のステップでは、両方のスプレッド シートを 1 つのワークブックへ結合します。上記のように作成した両方のスプレッド シートを、開く必要があります。
2. 上記のように作成したスプレッド シート temp.xls に切り替えます。
3. [Sheet 1] タブ (スプレッド シートの下にある) で右クリックし、[移動またはコピー...] をクリックします。



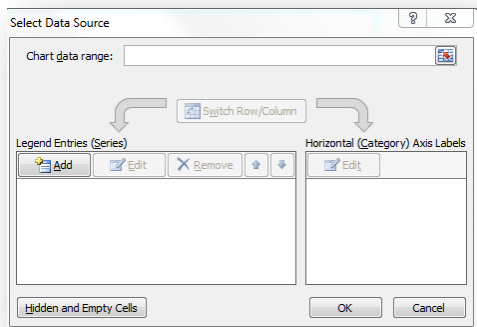
4. 最初に作成した Excel 文書にシートを移動します。



5. temp.xls を閉じます。
6. 上記のように最初に作成し、2 つのシートを含んでいるスプレッド シートに切り替えます。Process¥Private Bytes クエリからの結果と、CLR Memory¥# Bytes in all Heaps クエリからの別の結果が 1 つになります。
7. 複数のログ ファイルの情報がデータに含まれている場合、両方のシートを ([データ] タブを用いて) [日時] 列の順にソートしてください。そうしないと、結果のグラフがおかしくなります。
8. [挿入] タブで、[グラフ]、[折れ線]、[2-D 折れ線] の順にクリックします。

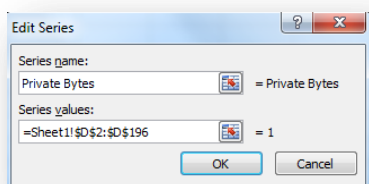


9. [データ] グループで、[データの選択] をクリックします。シートで選択されている部分があれば、その部分がグラフ化されます。すべての系列を削除することでグラフをクリアし、グラフのデータ範囲をクリアしてください。[データソースの選択] ダイアログ ボックスが表示されます。

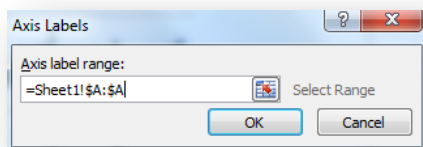
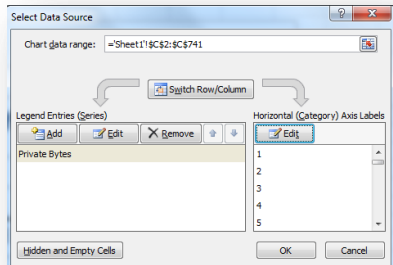


10. [追加] をクリックします。

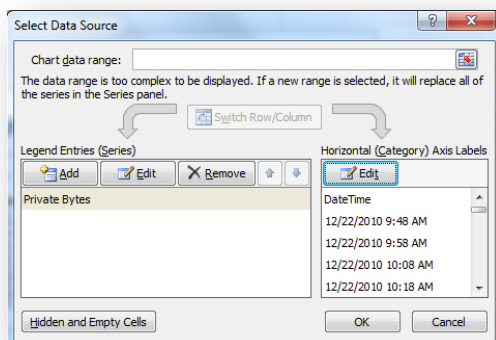
11. 最初の系列を「**Private Bytes**」系列と名付け、[バイト] 列からデータ範囲を選択します。



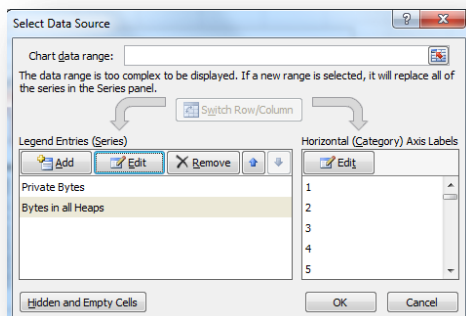
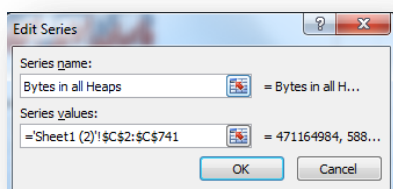
12. 次に、[日時] 列 (列 A) に横軸 (x-軸) ラベルを設定します。[編集] をクリックし、[Sheet 1] から列 A を選択します:



13. [OK] をクリックします。

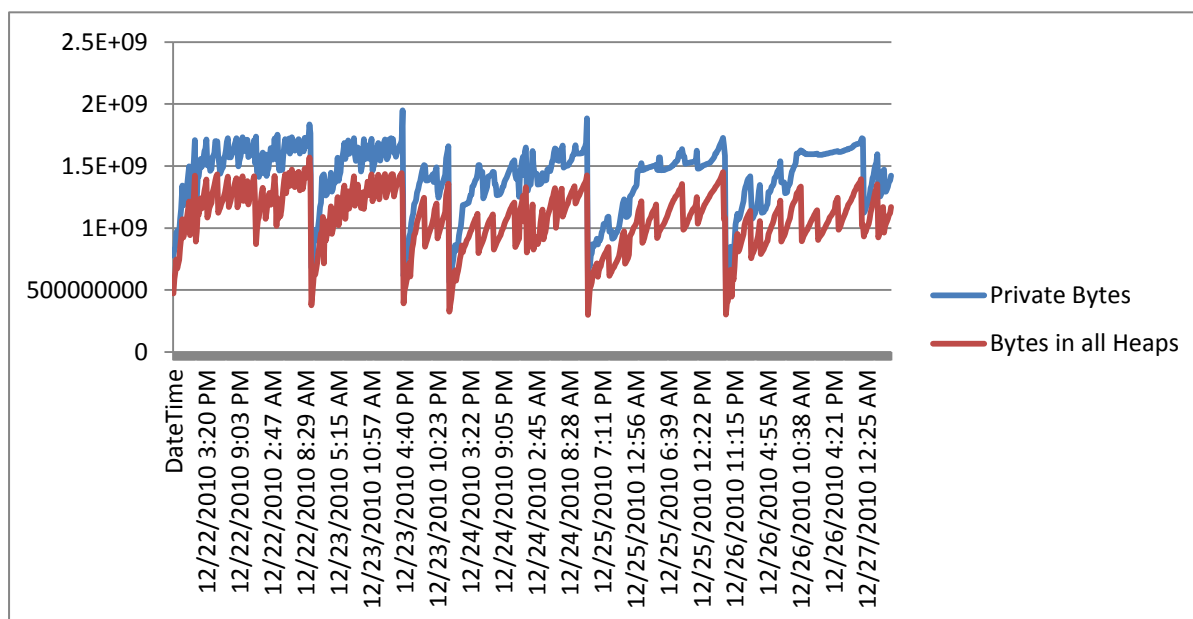


14. **[追加]** クリックして、2 番目の系列 ([Bytes in all Heaps]) に同じ操作をします。[バイト] 列からデータ範囲を選択します。[Bytes in all Heaps] の結果を含むシートのバイト列から、データ範囲を選択します。[Bytes in all Heaps] の横軸 (x-軸) に [Sheet 1] の列 A を設定する必要があることにもご注意ください。



15. **[OK]** をクリックします。

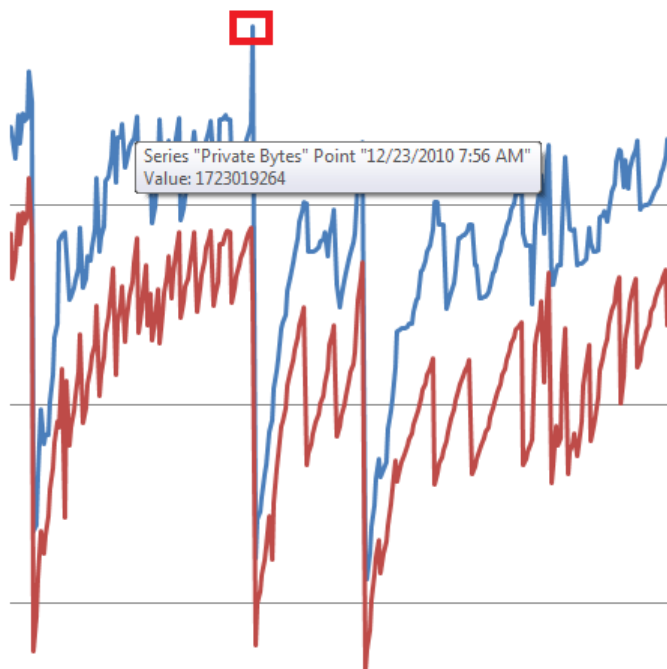
2.4.5 結果 (グラフ) の理解



グラフ 1

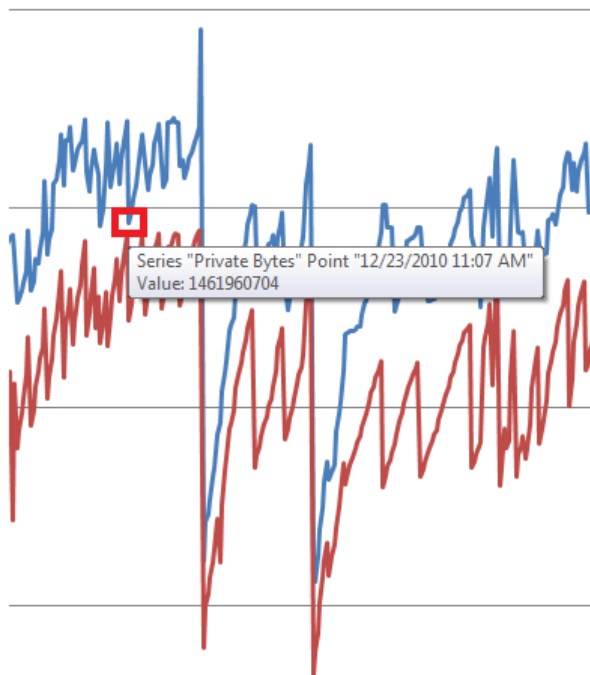
- このグラフは、マネージ メモリー スペースからのリークの例を示します。5 つある大きな下落箇所は、アプリケーション プールの再起動を表し、これはアプリケーション プールのプライベート メモリの限界値に達したか、OOM (メモリー不足) 例外が生じたか、どちらかの原因で起こります。下落と下落の間で、Private bytes と Heap bytes が平行して増加する傾向にあることがわかります。これは、メモリー リークの症状です。
- このサンプルでは、Private Bytes および Bytes in all Heaps の両方が同じ割合で増減していることがわかります。メモリーが安定点まで増加、そこにとどまるのことは、問題ありません。ただし、このサンプル グラフでは、メモリーがリセットされたときの 5 つのイベントを示していることにご注意ください。詳細な調査については、後で説明します。
- Bytes in all Heaps が一定のままの間、Private Bytes が増加していた場合、アンマネージ メモリー スペースからのリークの可能性を意味することがあり、詳細な調査が必要になります。

- ピークは、メモリの割り当てを示します。マウスをピーク上で移動すると、割り当てられたメモリー量、およびイベントの時刻についての情報が示されます。



グラフ 2

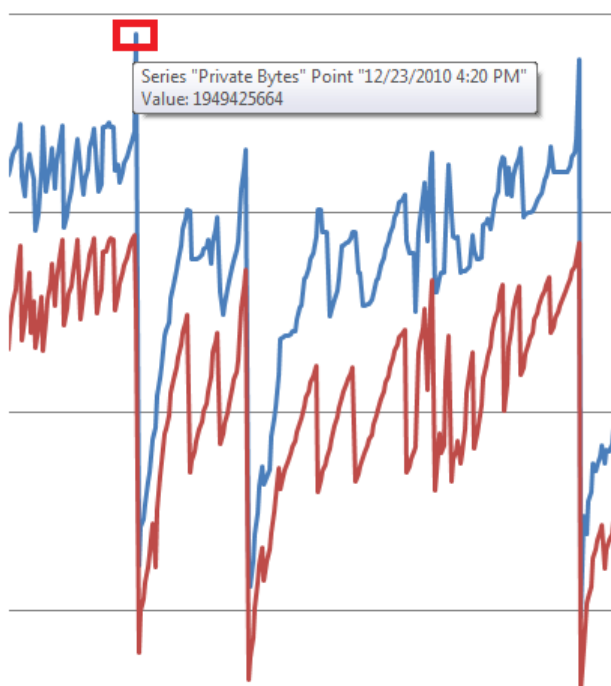
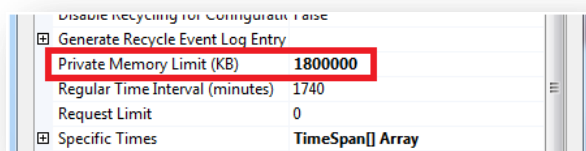
- 詳細に見ると、Private Bytes の急な山形は、アプリケーション プールのリサイクル、または OOM 例外より前の、メモリー使用のピークを示します。
- 谷間は、ガベージ コレクション イベントが生じた時を示します。マウスを谷間上で移動すると、まだ割り当てられているメモリー、およびイベントが生じた時刻についての情報が示されます。



グラフ 3

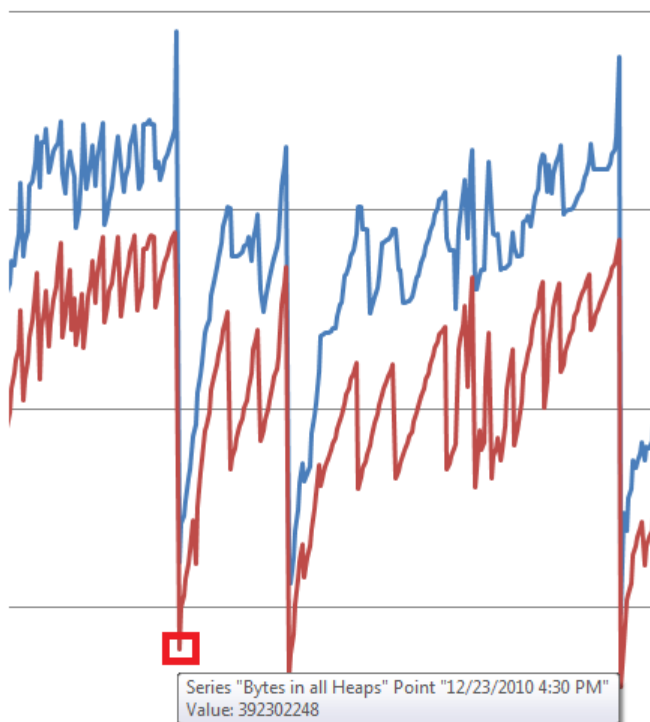
- 谷間、すなわち GC イベントの詳細これは、アプリケーション プールのリサイクルや OOM 例外ではなく、ガベージ コレクターの正常なアクティビティであることにご注意ください。
- 最も高いピークと最も低い谷間を見て得られた情報を、Windows イベント ログから得られる情報と関連付けることができます。

- メモリーがリセットされる直前の高い点は、アプリケーション プール用に設定されたプライベート メモリーの限界値と比較することができます (IIS マネージャーに進み、適切なアプリケーション プールを選択し、詳細設定を選択します)。ご注意ください: アプリケーション プールでプライベート メモリーの限界値が設定されていない場合、複数の OutOfMemory 例外が同じ時間枠の間に生じているか Sitecore ログを確認します(下の図から、メモリーが 1800000 KB に達したときに、アプリケーション プールが設定されることがわかります)。



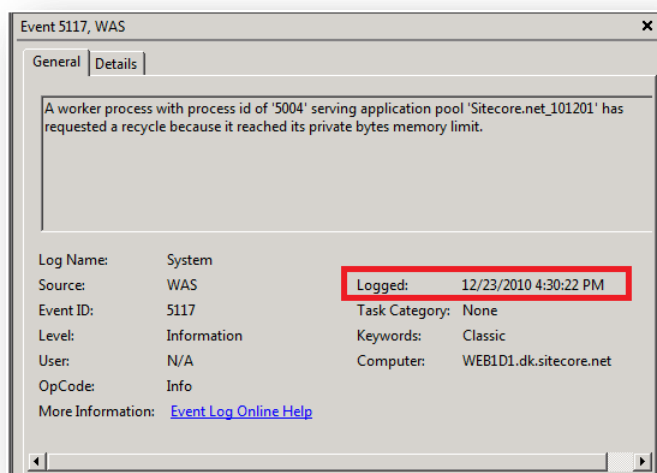
グラフ 4

- アプリケーション プールのリサイクル、または OOM 例外の直前の、メモリーの急な山形についての詳細情報は、メモリーの使用がアプリケーション プールで使用可能なメモリーを超えるかどうかを確認するために使用することができます。
- ピークの次にくる低い点と、Windows イベント ログの情報とを比較すると、アプリケーション プールのリサイクルが実際に起こったことがわかります。Private Bytes および Bytes in all Heaps の両方で、こうして一定割合で増加した後に、メモリーが限界値を超えたり、アプリケーション プールのリサイクルが起こったりします。こうした増加によって、マネージ メモリー スペースからのリークが生じる可能性があります。詳細な調査が必要となります。



グラフ 5

- メモリー リセット イベントの詳細について。使用可能な日時を用いて、Sitecore ログと Windows イベント ログから入手可能な情報を関連付けて、OOM 例外および/またはアプリケーション プールのリサイクル イベントが起きたかどうか確認することができます。



(タイムスタンプの一致は、アプリケーション プールがリサイクルされたことを示します)

2.4.6 メモ

.net メモリの使用およびリークの調査についての追加情報は、次のサイトで説明されています:

- <http://msdn.microsoft.com/ja-jp/magazine/cc163491.aspx>
- [http://msdn.microsoft.com/en-us/library/Ee817660\(pandp.10\).aspx](http://msdn.microsoft.com/en-us/library/Ee817660(pandp.10).aspx)

第 3 章

チューニング手順 - 全般

チューニング手順 - 全般は、Sitecore の実装が最高のパフォーマンスで動作するように構成されているかどうかを確認するために設計された一連のタスクです。

それぞれのタスクには、入門情報、必要なスキル、起こりうる症状、設定を確認する手順、問題の解決方法、記録する結果についての情報が含まれます。

より具体的なチューニング手順は、次の章で説明します。

この章には次のセクションがあります。

- SQL Server インデックスの断片化レベルのチェック
- SQL Server メンテナンス プランのチェック
- データベース テーブルのクリーンアップ
- データベース クリーンアップ エージェントのチェック
- 未使用の検索インデックスの無効化
- ソフトウェアおよびサーバーの設定

3.1 SQL Server インデックスの断片化レベルのチェック

インデックスが古くなるにつれ、連続しないデータの挿入と削除が負担となって、断片化を生じる原因となる可能性があります。頻繁に使用される CMS データベースでは、ほんの数日間でこの現象が起こることがあります。少量の断片化であれば、通常はパフォーマンスに悪影響を及ぼしません。しかし、断片化の割合が上がるにつれ、パフォーマンスは劇的に影響を受けます。

3.1.1 必要なスキル

- SQL Server 2008 Management Studio の実務知識。

3.1.2 症状

- CPU 使用の劇的な増加。
- クエリにおけるパフォーマンスの低下。
- データベース書き込みにおけるパフォーマンスの低下。
- データベース サーバーへの接続の中断。
- 時間のかかるレンダリング。
- 時間のかかる Sitecore クライアント ツール — デスクトップ、コンテンツ エディターなど。

3.1.3 断片化されたインデックスのチェック手順

インデックス断片化の割合をチェックするために、CMS データベース (コア、マスター、Web) に対して物理統計の標準レポートのインデックスを、次のように実行します：

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** の *Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、**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. **[OK]** をクリックします。
5. **オブジェクト エクスプローラ**の *Master* データベースを右クリックし、**[レポート]**、**[標準レポート]**、**[インデックスの物理統計]** をクリックします。
6. SQL Server Management Studio は、テーブル名、インデックス名、インデックスの種類、パーティション数、および推奨されている**操作についての情報を表示する**レポートを生成します。
7. コアおよび *Web* データベースに関してはステップ 5 - 6 を繰り返し、インデックスに生じた断片化のレベルをチェックします。

3.1.4 結果の理解

出力はこのように見えます:

Index Name	Index Type	# Partitions	Depth	Operation Recommended
ndxID	NONCLUSTERED INDEX	1	2	Rebuild
ndxName	NONCLUSTERED INDEX	1	2	Rebuild
ndxParentID	NONCLUSTERED INDEX	1	2	Rebuild
ndxTemplateID	NONCLUSTERED INDEX	1	2	Rebuild

レポートに表示されるキー値の 1 つに、[推奨されている操作] フィールドがあります。いずれの再構築という値も、インデックスが断片化されていることを示します。

[# パーティション] フィールドを展開することで、所定のインデックスの断片化された割合 (%) を表示することができます。

Index Name	Index Type	# Partitions	Depth	Operation Recommended			
ndxID	NONCLUSTERED INDEX	1	2	Rebuild			
					Partition No.	Avg. Fragmentation (%)	# Fragments
			1	96	24	1	24
ndxName	NONCLUSTERED INDEX	1	2	Rebuild			

3.1.5 推奨事項

Sitecore は、インデックスの断片化を 10% 未満に保つことを推奨しています。

3.1.6 解決方法

CMS データベース (コア、マスター、Web) のインデックスをデフラグするために、デフラグのメンテナンス プランを次のように実行します:

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**で、[管理/メンテナンス プラン] フォルダーを展開します。
3. ***[インデックス デフラグのメンテナンス プラン]** を右クリックして、**[実行]** をクリックします。

このようなメンテナンス プランが存在しない場合は、「SQL Server メンテナンス プランのチェック」のセクションを参照してください。

調査結果レポート

結果の記録

インデックスの断片化の割合が 10% より大きいものがありますか __はい __いいえ

ページの断片化が 10% 未満です:

OK。インデックスの断片化検査結果から、CMS データベースのインデックスは、パフォーマンスに影響を与える程断片化されていないことが示されています。

ページの断片化が 10% を超えています:

エラー。インデックス断片化の検査結果から、CMS データベースのインデックスは断片化されていることが示されています。推奨事項では、データベースを最初にバックアップしてから、デフラグすることになっています。

3.2 SQL Server メンテナンス プランのチェック

メンテナンス プランによって、自動化されたタスク セットをスケジュールに従って実行することで、データベースのマニュアルメンテナンスの必要性が取り除かれます。このプランでは、データベースに対し定期的にチェックとメンテナンスを行い、データベースを最適の状態に保ちます。

3.2.1 必要なスキル

- SQL Server 2008 Management Studio の実務知識
- T-SQL スクリプトの実行の実務知識

3.2.2 症状

- データベースが最適のパフォーマンスまで達していない。
- インデックスが断片化されつつあり、対処が必要。

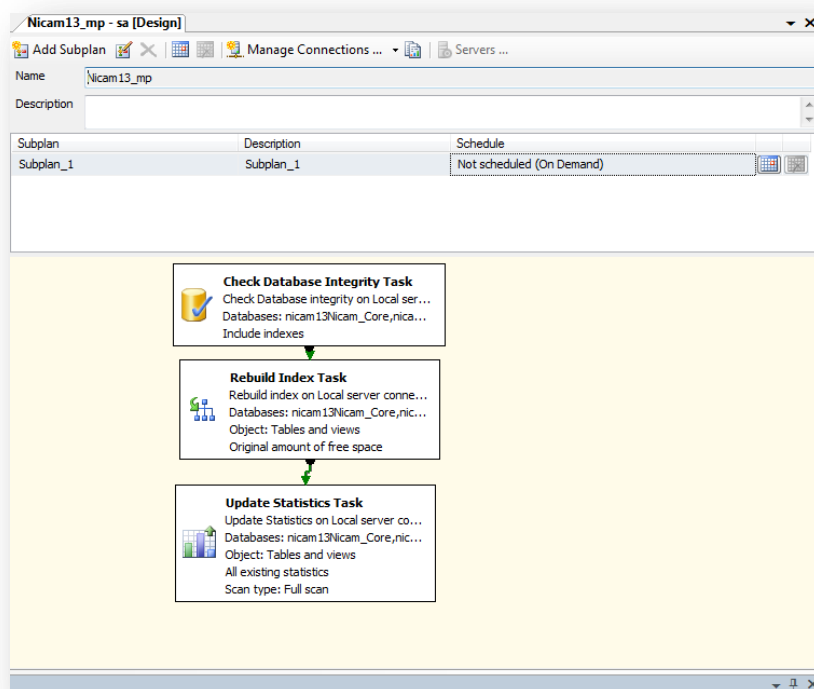
3.2.3 SQL Server メンテナンス プランのチェック手順

SQL Server メンテナンス プランが存在していて、かつベスト プラクティスに従っていることを確認するには、次を実行してください:

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**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、ダブルクリックしてどのように構成されているか確認します (このタスクの「調査結果」セクションで使用されます)。

3.2.4 結果の理解

出力はこのように見えます:



メンテナンス プランには、以下のタスクが含まれます:

- 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
- 統計の更新タスク

3.2.5 推奨事項

Sitecore は、CMS データベースに SQL Server メンテナンス プランを設定する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、および統計の更新タスクが含まれます。

3.2.6 解決方法

SQL Server Management Studio は、メンテナンス プランの作成を単純化するために IDE を提供します。インデックスをデフラグするために、MP を作成します:

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**で、**[管理]** フォルダを展開します。
3. **[メンテナンス プラン]** フォルダで右クリックし、**[新しいメンテナンス プラン]** を選択します。

4. MP に「*Defragment CMS Indexes (CMS インデックスのデフラグ)*」のような意味のある名前を付けます。
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]**、**[統計の更新タスク]** をドラッグ アンド ドロップし、縦に同じ順番で並べます。
6. ボックスから別のボックスへの矢印をドラッグすることでタスクをつなげ、次のように接続します: **[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]** を右クリックして、**[編集]** を選択します。
8. **[接続]** および CMS データベース (コア、マスター、Web) を選択して、**[OK]** をクリックします。
9. **[インデックスの再構築タスク]** を右クリックして、**[編集]** を選択します。
10. **[接続]** および CMS データベース (コア、マスター、Web) を選択し、**[インデックスの再作成中にオンラインのインデックスを保持する]** チェック ボックスを選択して (SQL Server Enterprise Edition の場合のみ)、**[OK]** をクリックしてください。
11. **[統計の更新タスク]** を右クリックし、**[編集]** を選択します。
12. **[接続]** および CMS データベース (コア、マスター、Web) を選択し、オブジェクトに **[テーブルとビュー]**、**[すべての既存の統計]**、および **[スキャンの種類]** として **[フル スキャン]** を設定し、**[OK]** をクリックします。
13. **[スケジュール]** アイコン (右上の隅) の横の**[カレンダー]** アイコンをクリックし、スケジュールを毎週実行するように設定します。
14. **保存** します。

調査結果レポート

結果の記録

メンテナンス プランが作成されていますか。__ はい __ いいえ

メンテナンス プランがアナリティクス データベースに対して設定されていますか。__ はい __ いいえ

MP はスケジューリングされていますか、または、要求に応じて実行されますか。__ スケジュール __ 必要に応じて
スケジューリングされている場合、どのような間隔で実行されますか。 _____

MP はどのような手順を実行しますか。

メンテナンス プランが作成されており、以下の「調査結果」に記述したベスト プラクティスに従っています。

- メンテナンス プランが作成されていますか。 **はい**
- メンテナンス プランがアナリティクス データベースに対して設定されていますか。 **はい**
- MP はスケジューリングされていますか、または、要求に応じて実行されますか。 **—スケジュール**
- スケジューリングされている場合、どのような間隔で実行されますか。 **—毎日**
- MP はどのような手順を実行しますか。 - データベースの整合性の確認、インデックスの再構築タスク、統計情報の更新

OK。SQL Server メンテナンス プランのチェック結果により、メンテナンス プランが作成、適切にスケジューリングされており、Sitecore のベスト プラクティスに従っていることが分かりました。

メンテナンス プランが作成されていますが、ベスト プラクティスに従っていません。

エラー。SQL Server メンテナンス プランのチェック結果により、メンテナンス プランが作成されていますが、Sitecore のベスト プラクティスに従っていないことが分かりました。

メンテナンス プランが作成されていません。

エラー。SQL Server メンテナンス プランのチェック結果により、現在メンテナンス プランが作成されていないことが分かりました。

3.2.7 メモとコメント

このタスクでは、Core、Master、Web の各データベースのインデックスを再構築する単一のメンテナンス プランの作成、および毎週メンテナンス プランを実行するスケジューリングの設定について説明します。いくつかのデータベースのインデックスのフラグメンテーションが他のデータベースより進んでいる場合、Core、Master、Web の各データベースでメンテナンス プランを分割します。これにより、インデックスのフラグメンテーションの進度に従い異なるスケジューリングを設定できます。

3.3 データベース テーブルのクリーンアップ

`web.config` ファイルには、`History`、`PublishQueue` および `EventQueue` テーブルからアーティファクト データを削除する一連のクリーンアップ エージェントが設定されています。しかし、時にはこれらのテーブルが大きくなりすぎて、エージェントの実行中にタイムアウトが発生する場合があります。

このタスクはテーブル サイズを確認するように設計されており、1000 行以上のテーブルの場合は切り捨てられます。

3.3.1 必要なスキル

- SQL Management Studio の実務知識。
- SQL スクリプト実行の実務知識。

3.3.2 症状

- 公開時のパフォーマンス低下。
- インデックス作成時のパフォーマンス低下。

3.3.3 History、PublishQueue、EventQueue テーブルのクリーンアップ手順

`History`、`PublishQueue`、`EventQueue` テーブルのクリーンアップ方法:

1. SQL Server Management Studio を起動します。
2. **新規クエリ** をクリックします。
3. 次のクエリを実行します。実行する際、`/* database name */` を、スクリプトを実行するデータベース名と置き換えます。

```
USE /* database name */

/* TRUNCATE History TABLE */
IF OBJECT_ID('History','U')ISNOTNULL
IF((SELECT COUNT(*) FROM [History])> 1000)
BEGIN
    TRUNCATE TABLE [History];
    PRINT 'Truncated the History Table';
END

/* TRUNCATE EventQueue TABLE */
IF OBJECT_ID('EventQueue','U')ISNOTNULL
if((SELECT COUNT(*) FROM [EventQueue])> 1000)
BEGIN
```

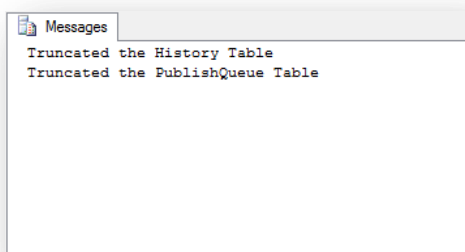


```
TRUNCATE TABLE [EventQueue];
PRINT 'Truncated the EventQueue Table';
END

/* TRUNCATE PublishQueue TABLE */
IF OBJECT_ID('PublishQueue','U')ISNOTNULL
IF((SELECT COUNT(*) FROM [PublishQueue])> 1000)
BEGIN
    TRUNCATE TABLE [PublishQueue];
    PRINT 'Truncated the PublishQueue Table';
END
```

3.3.4 結果の理解

出力結果は、次のようなものとなります:



- [メッセージ] ウィンドウに、切り捨てられたテーブル一覧が表示されます。

3.3.5 推奨事項

Sitecore は、History、PublishQueue、EventQueue テーブルの行数 (エントリ数) を 1000 より少なくすることを推奨しています。これにより、クリーンアップ エージェント実行中のタイムアウトを防ぐことができます。

3.3.6 解決方法

前述の手順を実行します。

調査結果レポート

結果の記録

History、*PublishQueue* または *EventQueue* テーブルのいずれかを切り捨てる必要があります: はい ___ いいえ _

—

History、*PublishQueue* または *EventQueue* テーブルのいずれかを切り捨てる必要があります: = いいえ

OK。 *History*、*PublishQueue* および *EventQueue* テーブルの行数は、推奨プラクティスどおりそれぞれ 1000 行より少ない行数です。

History、*PublishQueue* または *EventQueue* テーブルのいずれかを切り捨てる必要があります: = はい

エラー。 *History*、*PublishQueue* または *EventQueue* テーブルの行数は 1000 行以上あります。Sitecore の推奨では、*History*、*PublishQueue* および *EventQueue* テーブルは 1000 行より少ない行数を推奨しています。

3.4 データベース クリーンアップ エージェントのチェック

`web.config` ファイルには、`History`、`PublishQueue` および `EventQueue` テーブルからアーティファクト データを削除する一連のクリーンアップ エージェントが設定されています。

このタスクは、それらのタスクの設定および定期的なスケジューリングによる実行をチェックするよう設計されています。

3.4.1 必要なスキル

- Sitecoreweb.config ファイルの実務知識。

3.4.2 症状

- 公開時のパフォーマンス低下。
- インデックス作成時のパフォーマンス低下。

3.4.3 History、PublishQueue、EventQueue クリーンアップ タスクの設定 チェック手順

クリーンアップ タスクが定期的に実行されるよう設定されているかを確認するにはどこを参照すればよいかを示す手順は次のとおりです：

1. お使いのエディターで、`web.config` ファイルを開きます。
2. `<configuration><sitecore><scheduling></scheduling>` ノードに移動します。

3.4.4 結果の理解

出力結果は次のようなものとなります (分かりやすくするために、`<agent>` ノードはいくつか削除しています)：

```
<scheduling>
<!-- Time between checking for scheduled tasks waiting to execute -->
<frequency>00:05:00</frequency>

<!-- Agent to clean up history data -->
<agenttype="Sitecore.Tasks.CleanupHistory"method="Run"interval="04:00:00"/>
<!-- Agent to clean up publishing queue -->
<agenttype="Sitecore.Tasks.CleanupPublishQueue, Sitecore.Kernel"method="Run"interval="04:00:00"
>

<DaysToKeep>30</DaysToKeep>
</agent>
<!-- Agent to clean up the event queue -->
<agenttype="Sitecore.Tasks.CleanupEventQueue, Sitecore.Kernel"method="Run"interval="04:00:00">
<DaysToKeep>1</DaysToKeep>
</agent>
```

- エージェントの実行準備が整っているかの確認のために、<scheduling><frequency> に **00:00:00** 以外の値を設定します。
- *Sitecore.Tasks.CleanupHistory* の間隔が **00:00:00** より大きい値になっていることを確認します。
- *Sitecore.Tasks.CleanupPublishQueue* の間隔が **00:00:00** より大きい値になっていることを確認します。
- *Sitecore.Tasks.CleanupEventQueue* の間隔が **00:00:00** より大きい値になっていることを確認します。

3.4.5 推奨事項

Sitecore は、値を **00:00:00** より大きい値に設定することでスケジューリングの頻度を有効にし、*History*、*PublishQueue* および *EventQueue* のクリーンアップ エージェントの間隔に **00:00:00** より大きな値を設定することを推奨します。(頻度のデフォルト値は **00:05:00** であり、間隔のデフォルト値は **04:00:00** です。)

3.4.6 解決方法

スケジューリング頻度、*Sitecore.Tasks.CleanupHistory* の間隔、*Sitecore.Tasks.CleanupPublishQueue* の間隔、および/または *Sitecore.Tasks.CleanupEventQueue* が **00:00:00** に設定されている場合、実行するには値を大きくする必要があります。未変更の *web.config* ファイルで指定しているデフォルト値は次のとおりです:

- スケジューリングの頻度 = **00:05:00** (5 分)
- *Sitecore.Tasks.CleanupHistory* 間隔 = **04:00:00** (4 時間)
- *Sitecore.Tasks.CleanupPublishQueue* 間隔 = **04:00:00** (4 時間)
- *Sitecore.Tasks.CleanupEventQueue* 間隔 = **04:00:00** (4 時間)

調査結果レポート

結果の記録

スケジューリングの頻度は 00:00:00 より大きい はい ___ いいえ ___

スケジューリングの頻度は 00:00:00 より大きい はい ___ いいえ ___

Sitecore.Tasks.CleanupPublishQueue の間隔は 00:00:00 より大きい はい ___ いいえ ___

Sitecore.Tasks.CleanupEventQueue の間隔は 00:00:00 より大きい はい ___ いいえ ___

スケジューリングの頻度は 00:00:00 より大きい = はい:

OK。スケジューリングの頻度は 00:00:00 より大きい値に設定されているため、エージェントは指定された間隔で実行されます。

スケジューリングの頻度は 00:00:00 より大きい = いいえ:

エラー。スケジューリングの頻度は 00:00:00 に設定されているため、エージェントは指定された間隔で実行されません。Sitecore は、スケジューリングの頻度を 00:00:00 より大きい値に設定することを推奨しています。

Sitecore.Tasks.CleanupHistory の間隔は 00:00:00 より大きい = はい:

OK。Sitecore.Tasks.CleanupHistory エージェントの間隔は 00:00:00 より大きい値に設定されているため、エージェントは定期的に行われます。

Sitecore.Tasks.CleanupHistory の間隔は 00:00:00 より大きい = いいえ:

エラー。Sitecore.Tasks.CleanupHistory の間隔は 00:00:00 に設定されているため、エージェントは定期的に行われません。Sitecore は、間隔を 00:00:00 より大きい値に設定することを推奨しています。

Sitecore.Tasks.CleanupPublishQueue の間隔は 00:00:00 より大きい = はい:

OK。Sitecore.Tasks.CleanupPublishQueue エージェントの間隔は 00:00:00 より大きい値に設定されてい

るため、エージェントは定期的に実行されます。

Sitecore.Tasks.CleanupPublishQueue の間隔は 00:00:00 より大きい = いいえ:

エラー。Sitecore.Tasks.CleanupPublishQueue の間隔は 00:00:00 に設定されているため、エージェントは定期的に実行されません。Sitecore は、間隔を 00:00:00 より大きい値に設定することを推奨しています。

Sitecore.Tasks.CleanupEventQueue の間隔は 00:00:00 より大きい = はい:

OK。Sitecore.Tasks.CleanupEventQueue エージェントの間隔は 00:00:00 より大きい値に設定されているため、エージェントは定期的に実行されます。

Sitecore.Tasks.CleanupEventQueue の間隔は 00:00:00 より大きい = いいえ:

エラー。Sitecore.Tasks.CleanupEventQueue の間隔は 00:00:00 に設定されているため、エージェントは定期的に実行されません。Sitecore は、間隔を 00:00:00 より大きい値に設定することを推奨しています。

3.5 未使用の検索インデックスの無効化

インデックスの再構築および/または更新作業は、多くのリソースを消費することがあります。使用されていない検索インデックスは更新すべきではありません。

このタスクでは、インデックス更新を無効にする方法を説明しています。

3.5.1 必要なスキル

- Sitecoreweb.config ファイルの実務知識。

3.5.2 症状

- インデックス作成時のパフォーマンス低下。

3.5.3 検索インデックスの更新の無効化手順

クリーンアップ タスクが定期的に行われるよう設定されているかを確認するにはどこを参照すればよいかを示す手順は次のとおりです:

1. お使いのエディターで、web.config ファイルを開きます。
2. Indexing.UpdateInterval の更新間隔を 00:00:00 <setting name="Indexing.UpdateInterval" value="00:00:00" />

3.5.4 推奨事項

Sitecore は、使用されていない検索インデックスには、Indexing.UpdateInterval の値を 00:00:00 に設定しインデックスの更新を無効にすることを推奨しています。

3.5.5 解決方法

検索インデックスの更新の無効化手順 セクションを参照してください。

調査結果レポート

結果の記録

検索インデックスが使用されています: __ はい __ いいえ

Index.UpdateInterval の値 = _____

検索インデックスを使用しています = はい:

検索インデックスを使用している場合、このタスクは適用されません。

検索インデックスを使用しています = いいえ **および** Indexing.UpdateInterval = 00:00:00

OK。検索インデックスを使用しておらず、推奨プラクティスに従い更新間隔は 00:00:00 に設定されています。

検索インデックスを使用しています = いいえ **および** Indexing.UpdateInterval > 00:00:00

エラー。検索インデックスを使用しておらず、更新間隔が > 00:00:00 の場合、Sitecore は Indexing.UpdateInterval の値を 00:00:00 に設定し、インデックスの更新を無効にすることを推奨しています。

3.6 ソフトウェアおよびサーバーの設定

このチェックは、ソフトウェア (OS、IIS および SQL Server) およびサーバー設定が、パフォーマンスを最適化するための推奨プラクティスに従っているかどうかを確認します。サーバーのパフォーマンス レベルの記録には、元となるベースラインから開始する加点システムを使用します。スコアリングはセクションに分かれており、各セクションには独自の値が割り当てられています。どのセクションでも、得点 1 は Sitecore が推奨する最低限の要件を満たしていることを示しています。得点が 1 より大きい場合、より高いレベルで要件を満たしていることを示しています。得点が 1 より小さい場合、赤信号となり、サーバーのパフォーマンスが少なくとも Sitecore の推奨要件を満たすにはリソースの追加または変更が必要であることを示しています。

スコアリング システムは、SDN に示した (Sitecore バージョン 6.4.x をベースとした) インストール ガイドの最小要件に基づいています。

Sitecore の推奨するサーバー要件:

- すべてのサーバーは 64 ビット モードで稼働していなければなりません。
- すべてのサーバーは Windows Server 2008 R2 で、最新のセキュリティパッチを適用した状態で稼働していなければなりません。

Sitecore の推奨するコンテンツ オーサリング環境 (Web サーバー):

- IIS 7.x
- Quad Core プロセッサ
- 8 GB のメモリー
- 250 GB

Sitecore の推奨するコンテンツ オーサリング環境 (SQL Server):

- SQL Server 2008 R2 x64
- Dual Quad Core プロセッサ
- 12 GB のメモリー
- データ ファイル用の 250 GB のディスク
- ログ ファイル用の 250 GB のディスク

Sitecore の推奨する本番環境 (Web サーバー):

- IIS 7.x
- Dual Quad Core プロセッサ
- 8 GB のメモリー

- 250 GB のディスク

Sitecore の推奨する本番環境 (SQL Server):

- SQL Server 2008 R2 x64
- Dual Quad Core プロセッサ
- 12 GB のメモリー
- データ ファイル用の 250 GB のディスク
- ログ ファイル用の 250 GB のディスク

Sitecore の推奨する Sitecore アナリティクス (OMS 1.x) 環境 (SQL Server):

- SQL Server 2008 R2 x64
- Dual Quad Core プロセッサ
- 16 GB のメモリー
- データ ファイル用の 500 GB のディスク
- ログ ファイル用の 250 GB のディスク
- 一時領域用の 250 GB のディスク

3.6.1 必要なスキル

- システム インフラストラクチャの実務知識。

3.6.2 症状

- 低いパフォーマンス。
- トラフィック増加に伴うパフォーマンス低下。

3.6.3 サーバー設定のスコアリング

サーバー、ソフトウェアおよびハードウェア、設定のスコアリングは、スコアリング表に分けられています。各表に記録されたスコアは、システムが Sitecore の推奨要件を満たしているか、満たしていないか、十分に満たしているかを決定します。

すべてのサーバーの一般要件/OS

Windows Server	2008 R2 x64	2008 x64	2003 x64	32 ビット OS
----------------	-------------	----------	----------	-----------

Windows Server	2008 R2 x64	2008 x64	2003 x64	32 ビット OS
	1	-1	-2	-4

スコア _____

コンテンツ オーサリング環境 (Web サーバー)

IIS バージョン	7.x	6.x	5.x
	+1	-1	-4

スコア _____

コア プロセッサ	8+	4	2	1
	+2	1	-1	-2

スコア _____

メモリー	16 GB 以上	12 GB	8 GB	4 GB	2 GB 以下
	+2	+1	1	-2	-4

スコア _____

コンテンツ オーサリング環境 (SQL Server)

SQL Server のバージョン	2008 R2	2008	2005

SQL Server のバージョン	2008 R2	2008	2005
	1	-1	-2

スコア _____

コア プロセッサ	12+	8	4	2
	+2	1	-1	-2

スコア _____

メモリー	16 GB 以上	12 GB	8 GB	4 GB	2 GB 以下
	+2	1	-1	-2	-4

スコア _____

本番環境 (Web サーバー)

IIS バージョン	7.x	6.x	5.x
	+1	-1	-4

スコア _____

コア プロセッサ	12+	8	4	2
	+2	1	-1	-2

スコア _____

メモリー	16 GB 以上	12 GB	8 GB	4 GB	2 GB 以下
	+2	+1	1	-2	-4

スコア _____

本番環境 (SQL Server)

SQL Server のバージョン	2008 R2	2008	2005
	1	-1	-2

スコア _____

コア プロセッサ	12+	8	4	2
	+2	1	-1	-2

スコア _____

メモリー	16 GB 以上	12 GB	8 GB	4 GB	2 GB 以下
	+2	1	-1	-2	-4

スコア _____

Sitecore アナリティクス (OMS 1.x) 環境 (SQL Server)

SQL Server のバージョン	2008 R2	2008	2005
	1	-1	-2

スコア _____

コア プロセッサ	12+	8	4	2
	+2	1	-1	-2

スコア _____

メモリー	20 GB 以上	16 GB	12 GB	8 GB	4 GB 以下
	+2	1	-1	-2	-4

スコア _____

データ、ログ、一時領域用に別個のディスク	はい	いいえ
	1	-2

スコア _____

調査結果レポート

すべてのサーバーの一般要件/OS

スコア \geq 1

OK。インストールされている OS のバージョンは、Sitecore の推奨要件を満たしています。

スコア $<$ 1

エラー。インストールされている OS のバージョンは、Sitecore の推奨要件を満たしていません。Sitecore は、拡張機能とバグ問題の解消により、Microsoft Windows 2008 R2 の使用を強く推奨しています。

コンテンツ オーサリング環境 (Web サーバー) - IIS バージョン

スコア \geq 1

OK。稼働している IIS のバージョンは、Sitecore の推奨要件を満たしています。

スコア $<$ 1

エラー。稼働している IIS のバージョンは、Sitecore の推奨要件を満たしていません。Sitecore は IIS バージョン 7.x の使用を強く推奨しています。

コンテンツ オーサリング環境 (Web サーバー) - コア プロセッサ

スコア \geq 1

OK。サーバーにインストールされている CPU コアの本数は、4 コア プロセッサ以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされている CPU コア の数は、4 コア プロセッサ以上という Sitecore の推奨要件を満たしていません。

コンテンツ オーサリング環境 (Web サーバー) - メモリー

スコア >= 1

OK。サーバーにインストールされているメモリー サイズは、8 GB 以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされているメモリー サイズは、8 GB 以上という Sitecore の推奨要件を満たしていません。

コンテンツ オーサリング環境 (SQL Server) - SQL Server バージョン

スコア >= 1

OK。稼働している SQL Server のバージョンは、Sitecore の推奨要件を満たしています。

スコア < 1

エラー。稼働している SQL Server のバージョンは、Sitecore の推奨要件を満たしていません。Sitecore は SQL Server バージョン 2008 R2 x64 の使用を強く推奨しています。

コンテンツ オーサリング環境 (SQL Server) - コア プロセッサ

スコア ≥ 1

OK。サーバーにインストールされている CPU コア の数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされている CPU コア の数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしていません。

コンテンツ オーサリング環境 (SQL Server) - メモリー

スコア ≥ 1

OK。サーバーにインストールされているメモリー サイズは、12 GB 以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされているメモリー サイズは、12 GB 以上という Sitecore の推奨要件を満たしていません。

本番環境 (Web サーバー) - IIS バージョン

スコア ≥ 1

OK。稼働している IIS のバージョンは、Sitecore の推奨要件を満たしています。

スコア < 1

エラー。稼働している IIS のバージョンは、Sitecore の推奨要件を満たしていません。Sitecore は IIS バージョン 7.x の使用を強く推奨しています。

本番環境 (Web サーバー) - コア プロセッサ

スコア >= 1

OK。サーバーにインストールされている CPU コアの本数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされている CPU コアの本数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしていません。

本番環境 (Web サーバー) - メモリー

スコア >= 1

OK。サーバーにインストールされているメモリー サイズは、8 GB 以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされているメモリー サイズは、8 GB 以上という Sitecore の推奨要件を満たしていません。

本番環境 (SQL Server) - SQL Server バージョン

スコア ≥ 1

OK。稼働している SQL Server のバージョンは、Sitecore の推奨要件を満たしています。

スコア < 1

エラー。稼働している SQL Server のバージョンは、Sitecore の推奨要件を満たしていません。Sitecore は SQL Server バージョン 2008 R2 x64 の使用を強く推奨しています。

本番環境 (SQL Server) - コア プロセッサ

スコア ≥ 1

OK。サーバーにインストールされている CPU コア の数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされている CPU コア の数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしていません。

本番環境 (SQL Server) - メモリー

スコア ≥ 1

OK。サーバーにインストールされているメモリー サイズは、12 GB 以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされているメモリー サイズは、12 GB 以上という Sitecore の推奨要件を満たしていません。

Sitecore アナリティクス (OMS 1.x) 環境 (SQL Server) - SQL Server バージョン

スコア >= 1

OK。稼働している SQL Server のバージョンは、Sitecore の推奨要件を満たしています。

スコア < 1

エラー。稼働している SQL Server のバージョンは、Sitecore の推奨要件を満たしていません。Sitecore は SQL Server バージョン 2008 R2 x64 の使用を強く推奨しています。

Sitecore アナリティクス (OMS 1.x) 環境 (SQL Server) - コア プロセッサ

スコア >= 1

OK。サーバーにインストールされている CPU コアの本数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされている CPU コアの本数は、8 コア プロセッサ以上という Sitecore の推奨要件を満たしていません。

Sitecore アナリティクス (OMS 1.x) 環境 (SQL Server) - メモリー

スコア ≥ 1

OK。サーバーにインストールされているメモリー サイズは、16 GB 以上という Sitecore の推奨要件を満たしています。

スコア < 1

OK。サーバーにインストールされているメモリー サイズは、16 GB 以上という Sitecore の推奨要件を満たしていません。

Sitecore アナリティクス (OMS 1.x) 環境 (SQL Server) - ディスク

スコア ≥ 1

OK。データ、ログ、一時領域は別個のディスクに保存されており、Sitecore の推奨要件を満たしています。

スコア < 0

エラー。データ、ログ、一時領域は別個のディスクに保存されておらず、Sitecore の推奨要件を満たしていません。

第 4 章

チューニング手順 - データベースのプロパティ

チューニング手順 - データベースのプロパティには、SQL Server のデータベース プロパティの設定をチェックするために設計された一連のタスクが含まれています。これらのプロパティを適切に設定することにより、Sitecore を最善のパフォーマンスで実行できます。

この章には次のセクションがあります。

- SQL Server 2008 (100) に設定された互換性レベル
- 自動終了プロパティを false に設定
- 自動圧縮プロパティを false に設定
- 復旧モデルを単純に設定

4.1 SQL Server 2008 (100) に設定された互換性レベル

互換性レベルは SQL 構文およびクエリ解析に影響を与えます。また、パフォーマンスへ影響を与えてはなりません。互換性レベルを SQL Server 2008 (100) に設定することにより、CMS パフォーマンス チューニング ガイドで使用されている多くのスクリプト/ コマンドで使用されている、新しい T-SQL 機能を活用できます。

4.1.1 必要なスキル

- SQL Management Studio の実務知識。

4.1.2 症状

- パフォーマンスを向上するためのスクリプトが実行できません。

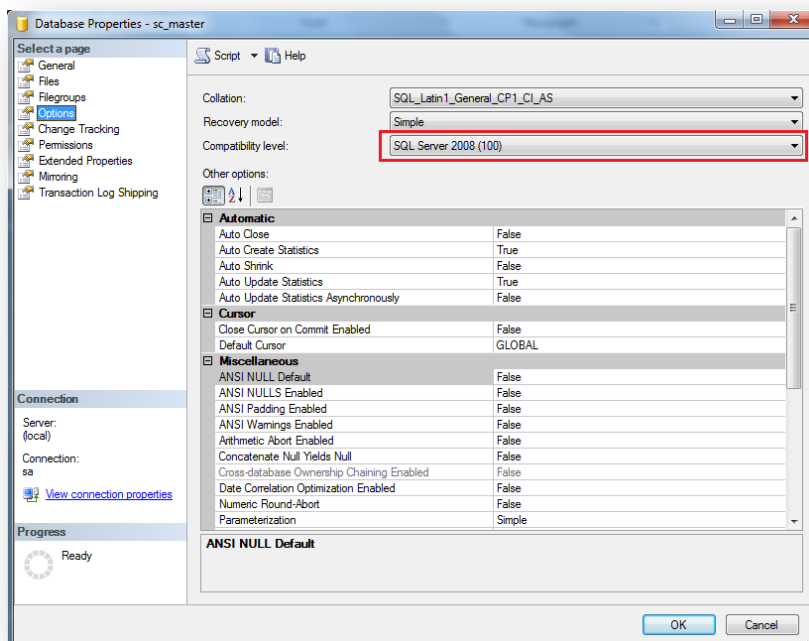
4.1.3 データベースの互換性レベルのチェック手順

データベースの互換性レベルの値をチェックする方法:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、互換性レベルのプロパティを参照します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

4.1.4 結果の理解

出力結果は、次のようなものとなります:



4.1.5 推奨事項

Sitecore は、**互換性レベル** プロパティの設定として **SQL Server 2008 (100)** を推奨しています。

4.1.6 解決方法

互換性レベル プロパティの設定方法は次のとおりです:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックして、**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. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

調査結果レポート

結果の記録

互換性レベル プロパティは SQL Server 2008 (100) に設定されています: __ はい __ いいえ

互換性レベル プロパティは **SQL Server 2008 (100)** に設定されています:

OK。互換性レベル プロパティは正しい値に設定されており、新しい T-SQL 機能を活用できます。

互換性レベル プロパティは **SQL Server 2008 (100)** に設定されていません:

エラー。互換性レベル プロパティは SQL Server 2008 (100) に設定されていません。Sitecore は、新しい T-SQL 機能を活用するために、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。

4.2 自動終了プロパティを false に設定

SQL Server がデータベースを開くと、その状態を維持するための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のためのメモリーがすべて割り当てられます。これらの操作には時間がかかります。自動終了プロパティは、これらのリソースをどのように処理するかを設定します。「true」または「ON」に設定されている場合、最後の接続が切断されるとリソースは解放されます。これは適切な処理のように思えますが、その後短時間で新しい接続が確立される場合 (1/10 秒以下)、これらすべてのリソースを再度割り当てる必要があります。自動終了プロパティを *false* または *OFF* に設定すると、この状態を回避できます。

4.2.1 必要なスキル

- SQL Server 2008 Management Studio の実務知識。

4.2.2 症状

- データベースへの接続に時間がかかります。

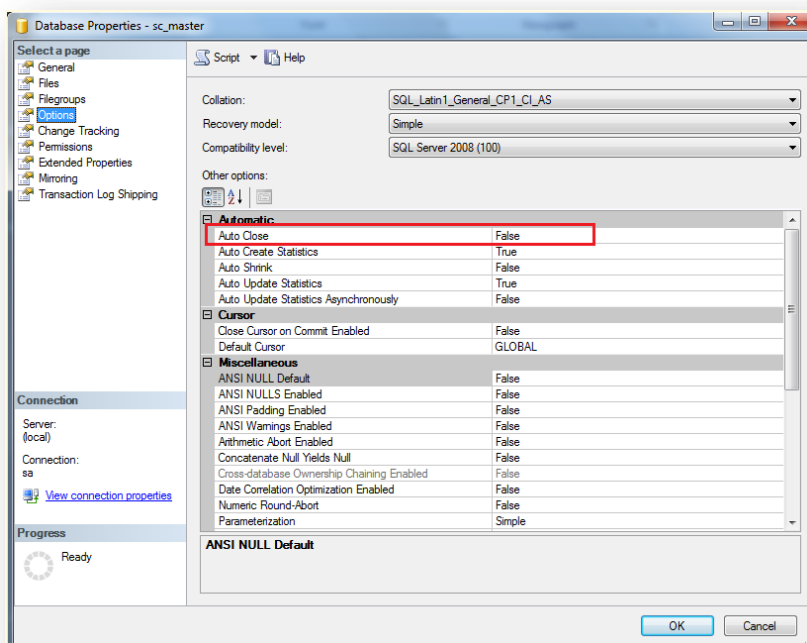
4.2.3 自動終了プロパティ値のチェック手順

自動終了プロパティ値のチェック方法:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、自動終了プロパティを参照します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

4.2.4 結果の理解

出力結果は、次のようなものとなります:



4.2.5 推奨事項

Sitecore は、**自動終了**プロパティを **false** に設定することを推奨しています。

4.2.6 解決方法

自動終了プロパティの設定方法は次のとおりです:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、自動終了プロパティが **false** に設定されていることを確認します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

調査結果レポート

結果の記録

自動終了プロパティは次のとおり設定されています: __ true __ false

自動終了プロパティは **false** に設定されています:

OK。自動終了プロパティは最善のパフォーマンスを得るための正しい値に設定されています。

自動終了プロパティは **true** に設定されています:

エラー。自動終了プロパティは最善のパフォーマンスを得るには正しくない値に設定されています。Sitecore は、自動終了プロパティを「false」に設定することを推奨しています。

4.3 自動圧縮プロパティを false に設定

自動圧縮プロパティには、a) コール時に大量のリソースを消費する、b) いつコールされるか制御できない、というデメリットがあります。自動圧縮と自動拡張を組み合わせると、データベースが継続的に圧縮と拡張を繰り返すというスパイラルに陥り、他のデータベースの重要なリソースを使ってしまうだけでなく、フラグメンテーションの問題も発生します。データベースまたはファイルで SHRINK コマンドを使用しなければならない場合、スクリプト、コマンドまたはスケジューリングされたメンテナンス プランから使用すべきです。自動圧縮プロパティを *false* または *OFF* に設定すると、この機能を無効にできます。

4.3.1 必要なスキル

- SQL Server 2008 Management Studio の実務知識。

4.3.2 症状

- パフォーマンスの低下。

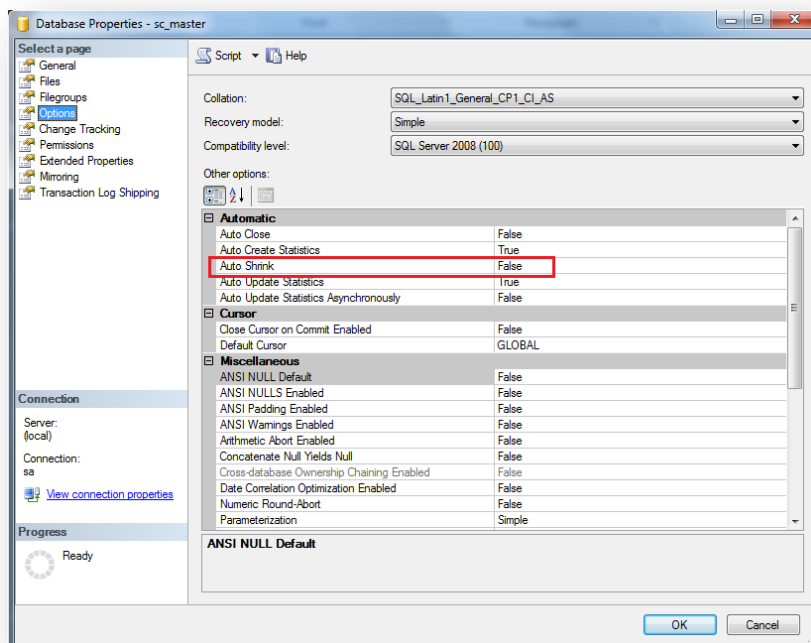
4.3.3 自動圧縮プロパティ値のチェック手順

自動圧縮プロパティ値のチェック方法:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、自動圧縮プロパティを参照します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

4.3.4 結果の理解

出力結果は、次のようなものとなります:



4.3.5 推奨事項

Sitecore は、**自動圧縮**プロパティを **false** に設定することを推奨しています。

4.3.6 解決方法

自動圧縮プロパティの設定方法は次のとおりです:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、自動圧縮プロパティが **false** に設定されていることを確認します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

調査結果レポート

結果の記録

自動圧縮プロパティは次のとおり設定されています: __ true __ false

自動圧縮プロパティは **false** に設定されています:

OK。自動圧縮プロパティは最善のパフォーマンスを得るための正しい値に設定されています。

自動圧縮プロパティは **true** に設定されています:

エラー。自動圧縮プロパティは最善のパフォーマンスを得るには正しくない値に設定されています。Sitecore は、自動圧縮プロパティを「false」に設定することを推奨しています。

4.4 復旧モデルを単純に設定

SQL Server の復旧モデルが単純に設定されている場合、トランザクション ログには最小限の情報を記録します。トランザクション ログが 70% 使用された場合、またはトランザクション ログのアクティブな箇所の復旧に、サーバー レベル設定の復旧間隔で指定した以上の時間がかかる場合、SQL Server はトランザクション ログを切り捨てます。

復旧モデルを単純に設定すると、完全または一括ログと比べて最小限の負荷で済みます。このことは、CMS データベースのパフォーマンス要件にとって非常に重要です。

4.4.1 必要なスキル

- SQL Server 2008 Management Studio の実務知識。

4.4.2 症状

- 復旧間隔時のパフォーマンスの低下。

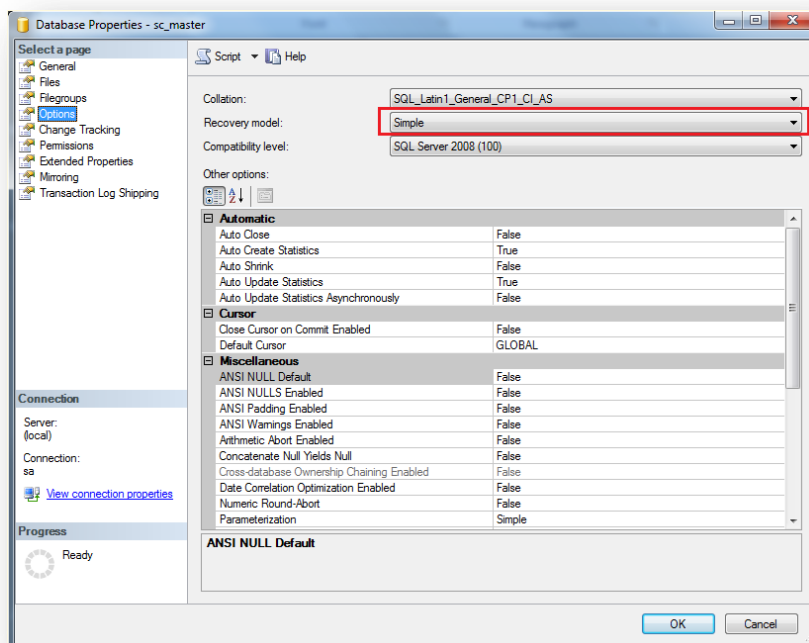
4.4.3 復旧モデル プロパティ値のチェック手順

復旧モデル プロパティ値のチェック方法:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、復旧モデル プロパティを参照します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

4.4.4 結果の解釈

出力結果は、次のようなものとなります:



4.4.5 推奨事項

Sitecore 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。

4.4.6 解決方法

復旧モデル プロパティの設定方法は次のとおりです:

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**で、*CMS Master* データベースを右クリックし、**[プロパティ]** をクリックします。
3. **[オプション]** ページをクリックし、復旧モデル プロパティが単純に設定されていることを確認します。
4. **[OK]** をクリックします。
5. *CMS Core* および *Web* データベースでも、手順 1 から 4 を繰り返します。

調査結果レポート

結果の記録

復旧モデル プロパティは**単純**に設定されています: ___ はい ___ いいえ

復旧モデル プロパティは**単純**に設定されています:

OK。復旧モデル プロパティは最善のデータベース パフォーマンスを得るための正しい値に設定されています。

復旧モデル プロパティは**単純**に設定されていません:

エラー。復旧モデル プロパティは**単純**に設定されていません。Sitecore は、最善のデータベース パフォーマンスを得るために復旧モデル プロパティを**単純**に設定することを推奨しています。

第 5 章

チューニング手順 - Sitecore のキャッシュ

Sitecore キャッシュの調査および設定は、複数のタスクに分割できます。これにより、各タスクを目的別に明確に分け、単純化できます。目的は、Sitecore データベース キャッシュ (プリフェッチ、データおよびアイテム キャッシュ) を設定、チューニングすることです。

出力レンダリング キャッシング プロパティの設定について、ユーザーはこれらのキャッシュを適切に有効にする方法およびキャッシュを期限切れにする方法の確認のために、Sitecore キャッシュ設定リファレンスおよび Sitecore プレゼンテーション コンポーネント リファレンスの両方を参照する必要があります。

Sitecore キャッシュの設定およびチューニングは、テスト環境または開発環境でのみ実施し、本番環境で実行すべきではありません。キャッシュを変更すると、保持されていたキャッシュが消去されてしまうためです。キャッシュが消去されると、本番環境でのユーザー エクスペリエンスが低下する可能性があります。

キャッシュのチューニングが完了後は、ピーク時を避けて、設定ファイルへの変更を本番環境に適用できます。

この章には次のセクションがあります。

- 初期キャッシュ値の設定
- Sitecore のキャッシュのチューニング
- プリフェッチ キャッシュ設定のガイドライン
- 出力 (レンダリング) キャッシュ設定のガイドライン

5.1 初期キャッシュ値の設定

使用する環境により、このタスクでは Sitecore データベースおよび html 出力キャッシュの初期値を設定できます。これらの値は、キャッシュのチューニング フェーズで使用されます。

注意

キャッシュ値の変更は、テスト環境または開発環境でのみ行うべきです。キャッシュのチューニングが完了後は、キャッシュへの変更を適用すると、キャッシュの内容が消去されるということを理解したうえで、新しいキャッシュ値を本番環境に適用できます。

5.1.1 必要なスキル

- Sitecore 設定ファイルの実務知識。

5.1.2 症状

- キャッシュに割り当てられるメモリの制限
- メモリ制限によるキャッシュからのデータの頻繁な消去
- 低いページ レンダリング パフォーマンス

5.1.3 初期値

次の表は、データベース キャッシュおよび html 出力キャッシュの推奨初期値を示しています。これらの値は、a) 環境と、b) 64 ビット モードで稼働していることに基づいています。32 ビット モードで稼働中のシステムについては、この半分の値を設定します。利用可能なメモリの制限により、32 ビット モードを使用しないことを強く推奨します。

環境/ターゲット	キャッシュ	値
コンテンツ配信のみ		
Web	プリフェッチ	200 MB
Web	データ	200 MB
Web	アイテム	200 MB
出力 (サイトごと)	html	100 MB
CMS のみ		

環境/ターゲット	キャッシュ	値
Master	プリフェッチ	200 MB
Master	データ	200 MB
Master	アイテム	200 MB
CMS およびコンテンツ配信が同じサーバーにある (Master および Web データベース)		
Master	プリフェッチ	200 MB
Master	データ	200 MB
Master	アイテム	200 MB
Web	プリフェッチ	150 MB
Web	データ	150 MB
Web	アイテム	150 MB
出力 (サイトごと)	html	100 MB
CMS およびコンテンツ配信が同じサーバーにある (ライブ モード)		
Master	プリフェッチ	300 MB
Master	データ	300 MB
Master	アイテム	300 MB
出力 (サイトごと)	html	100 MB

5.1.4 データ キャッシュおよびアイテム キャッシュを初期値に設定する手順

データ キャッシュおよびアイテム キャッシュの初期値への設定方法:

1. web.config ファイルをエディターで開きます (web の root ディレクトリにあります)。

2. <configuration><sitecore><databases><database id="x"><cacheSizes> に移動します (「x」は、上記の初期値表に示されたデータベースです)。

```
<cacheSizeshint="setting">
  <data>20MB</data>
  <items>10MB</items>
  <paths>500KB</paths>
  <standardValues>500KB</standardValues>
</cacheSizes>
```

3. データ キャッシュおよびアイテム キャッシュを初期値に設定します。
4. 実行環境に基づき、各データベースでこの手順を繰り返します。

5.1.5 プリフェッチ キャッシュを初期値に設定する手順

プリフェッチ キャッシュの初期値への設定方法:

1. プリフェッチ キャッシュの設定は、App_Config\Prefetch ディレクトリにあります。各データベース (Core、Master および Web) にはそれぞれ別個の設定ファイルがあります。上記の初期値表に基づき、作業しているデータベースのファイルを適切に編集します。
2. プリフェッチ設定ファイルを開きます。
3. <configuration><cacheSize> に移動します。

```
< c o n f i g u r a t i o n >
<cacheSize>20MB</cacheSize>
```

4. プリフェッチ キャッシュの初期値を設定します。
5. 実行環境に基づき、各データベースでこの手順を繰り返します。

5.1.6 html 出力キャッシュを初期値に設定する手順

データ キャッシュおよびアイテム キャッシュの初期値への設定方法:

1. web.config ファイルをエディターで開きます (web の root ディレクトリにあります)。
2. <configuration><sitecore><sites><site name="website"> に移動します。

```
<siteName="website"virtualFolder="/"physicalFolder="/"rootPath="/sitecore/content"startItem="/home"database="web"domain="extranet"allowDebug="true"cacheHtml="true"htmlCacheSize="100MB"registryCacheSize="0"viewStateCacheSize="0"xslCacheSize="5MB"filteredItemsCacheSize="2MB"enablePreview="true"enableWebEdit="true"enableDebugger="true"disableClientData="false"/>
```

3. html 出力キャッシュの初期値を設定します。
4. 実行環境に基づき、各ウェブサイトでこの手順を繰り返します。

5.1.7 推奨事項

Sitecore は、プリフェッチ、データ、アイテムおよび html 出力の各キャッシュを、パフォーマンス キャッシュのチューニングを行う前に初期値に設定することを推奨します。

5.1.8 解決方法

プリフェッチ、データ、アイテムおよび html 出力キャッシュを、お使いの環境に従った初期値に設定するには、上記の手順に従ってください。

調査結果レポート

結果の記録

プリフェッチ、データ、アイテムおよび html 出力の各キャッシュは、環境に従った初期値に設定されています ____ はい
____ いいえ

プリフェッチ、データ、アイテムおよび html 出力の各キャッシュは、環境に従った初期値に設定されています = はい

OK。この結果は、データベース キャッシュがチューニング前に設定済みであることを示しています。

プリフェッチ、データ、アイテムおよび html 出力の各キャッシュは、環境に従った初期値に設定されています = いいえ

エラー。この結果は、データベース キャッシュがチューニング前に設定済みでないことを示しています。Sitecore は、プリフェッチ、データ、アイテムおよび html 出力の各キャッシュを、パフォーマンス キャッシュのチューニングを行う前に初期値に設定することを推奨します。

5.2 Sitecore のキャッシュのチューニング

このタスクでは、負荷生成ツールを使用し、さまざまなキャッシュのサイズおよび消去の監視に Sitecore の cache.aspx ページを使用しながら、Sitecore のプリフェッチ、データ、アイテムの各キャッシュをチューニングする手順について説明しています。出力 (html) およびレンダリング キャッシュの詳細は、Sitecore キャッシュ設定リファレンスおよび Sitecore プレゼンテーション コンポーネント リファレンスマニュアルを参照してください。

キャッシュ チューニングは、時々実行する必要がある、継続的なプロセスです。コンテンツ/アイテムが追加されてウェブサイトが大きくなると、サイトを最適なパフォーマンス レベルで運営するためにキャッシュのサイズを変更しなければならない場合があります。

5.2.1 必要なスキル

- cache.aspx ページの実務知識。
- Sitecore 設定ファイルの実務知識。
- WCAT または Web Performance Load Tester のような負荷生成ツールの実務知識。

5.2.2 症状

- 低いリソリューション パフォーマンス。
- 高負荷のサーバー。
- システム容量の低下。

5.2.3 手順

Sitecore データベース キャッシュのチューニングは、負荷を生成してテスト サイトまたは開発サイトに負荷をかけ、キャッシュのサイズを最大サイズ (または初期値) と比較し、消去率を確認し、キャッシュ値を変更して再確認する、という手順を繰り返し実行することによって行われます。

5.2.4 必要条件

Sitecore データベース キャッシュは、チューニング前に初期値に設定する必要があります。次の項を参照してください。
初期キャッシュ値の設定

キャッシュの確認には、負荷生成ツールとスクリプトが必要です。Web 負荷を生成するプログラムの詳細は、WCAT を使用した負荷生成または Web Performance Load Tester の使用を参照してください。

(Sitecore の推奨) OS システム タイプは 64 ビット モードで稼働しています。

5.2.5 Sitecore データベース キャッシュのチューニング手順

1. すべての言語のすべてのアイテムがヒットするように、負荷を生成します。負荷生成の実行時間は、プロセス中の開始、停止が不要のように、チューニング手順全体を通じて実行されるようにしなければなりません。
2. 負荷生成が実行中に、/sitecore/admin/cache.aspx ページに移動します。環境に応じて、該当するキャッシュは Master (データ)、Master (アイテム)、Web (データ)、Web (アイテム)、SqlDataProvider - プリフェッチ データ (Master)、SqlDataProvider - プリフェッチ データ (Web) となります。

Name	Count	Size	Delta	MaxSize
------	-------	------	-------	---------

master[data]	6732	20MB	20MB	50MB
master[items]	9816	47MB	47MB	75MB

web[data]	0	0	0	20MB
web[items]	0	0	0	10MB

SqlDataProvider - Prefetch data(master)	7302	97MB	97MB	100MB
---	------	------	------	-------

3. キャッシュをチューニングするには、(ブラウザーの [再読み込み] ボタンではなく) ページの [更新] ボタンをクリックする必要があります。これにより、負荷に対してキャッシュがどのように動作するかを確認できます。最大サイズに比べて、キャッシュのサイズがどのようなサイズになっているか、デルタ値に伴って発生する変動がどうなっているかを確認する必要があります。次に、これに従ったキャッシュサイズ調整のガイドラインを示します (キャッシュに割り当てることができるメモリの最大サイズはシステムで利用可能なメモリの空き容量に応じて異なります。メモリの空き容量が多ければ多いほど、割り当てることができるサイズも大きくなります [ハードウェアの推奨タスクを参照してください]):
4. キャッシュ変動のデルタ値が一定である場合、またはキャッシュのサイズが常に最大サイズの 80% より大きい場合、キャッシュを 50% 増やします。
5. キャッシュのサイズが最大サイズの 50% 未満を保っている場合、メモリー消費を抑えるためにキャッシュサイズを 25% 減らします。
6. キャッシュが安定するまで、手順 3 を繰り返します。デルタの変動が一定で、キャッシュ サイズが 70% から 80% の間を保っているのが理想的です。

5.2.6 結果の理解

[cache.aspx ページのスクリーン ショットを挿入する]

5.2.7 推奨事項

Sitecore は、デルタの変動が一定で、キャッシュ サイズが 70% から 80% の間を保っている状態を推奨しています。

5.2.8 解決方法

上記の Sitecore データベース キャッシュのチューニング手順に従ってください。

調査結果レポート

結果の記録

Sitecore データベース キャッシュのチューニングが必要です ____ はい ____ いいえ

Sitecore データベース キャッシュのチューニングが必要です = いいえ

OK。この結果は、データベース キャッシュが適切にチューニングされており、Sitecore の推奨を満たしていることを示しています。

Sitecore データベース キャッシュのチューニングが必要です = はい

エラー。この結果は、データベース キャッシュがチューニング済みでないことを示しています。Sitecore は、Sitecore データベースのキャッシュ サイズがデルタの変動が一定で、キャッシュ サイズが 70% から 80% の間を保っている状態を推奨しています。

5.3 プリフェッチ キャッシュ設定のガイドライン

プリフェッチ キャッシュ設定のガイドラインはタスクではなく、プリフェッチ キャッシュについての入門、および設定、活用方法の推奨プラクティスの説明となります。

Sitecore キャッシングについて十分理解するには、Sitecore キャッシュ設定リファレンスを通読することを強く推奨します。次の項を参照してください。

http://sdn.sitecore.net/upload/sitecore6/sc62keywords/cache_configuration_reference_us.pdf

5.3.1 プリフェッチ キャッシュの理解

プリフェッチ キャッシュは、プリフェッチ設定ファイルが提供する情報に基づいてアプリケーションの初期化時に生成されます。これにより、アプリケーションの再起動後のユーザー エクスペリエンスがより円滑になります。ただし、プリフェッチ キャッシュの過度な利用は、アプリケーションの再起動に要する時間の増加につながり、ユーザー エクスペリエンスへのデメリットとなります。

アプリケーション起動後のプリフェッチ キャッシュの活用について理解するには、Sitecore キャッシングがどのように動作しているかを理解する必要があります：

データベース アイテム キャッシュにアイテムのエントリが含まれていない場合、Sitecore はデータベース データ キャッシュから該当エントリを取得し、別のタイプへと変換し、変換されたデータをエントリとしてデータベース アイテム キャッシュに保存します。データベース データ キャッシュにアイテムのエントリが含まれていない場合、Sitecore はデータベースのプリフェッチ キャッシュから該当エントリを取得し、別のタイプへと変換し、変換されたデータをエントリとしてデータベース データ キャッシュに保存します。エントリがデータベースのプリフェッチ キャッシュに含まれていない場合、Sitecore はデータベースのデータ プロバイダーからデータを取得し、別のタイプへと変換し、変換されたデータをエントリとしてデータベースのプリフェッチ キャッシュに保存します。

つまり、プリフェッチ キャッシュは初期化時に生成されるだけでなく、アプリケーション実行時にも生成されます。プリフェッチ キャッシュのサイズを考慮する際、この流れを理解する必要があります。

5.3.2 プリフェッチ キャッシュ設定ファイルの理解

プリフェッチ キャッシュ設定ファイルは、/App_Config/Prefetch ディレクトリにあります。データベースごとに別個のプリフェッチ キャッシュ設定ファイルがあり (Core.config, Master.config, Web.config)、/App_Config/Prefetch/Common.config ファイルと統合されています。

アイテム、テンプレートなどは、GUID によって設定ファイルで識別されています。

設定要素

- <configuration> - 設定ファイルのルート ノードです。

- `<cacheSize>` - プリフェッチ キャッシュの最大値を制御します。次はその例です: `<cacheSize>100MB</cacheSize>`.
- `<item>` - この要素は、アプリケーションの初期化時に、プリフェッチ キャッシュに読み込み特定のアイテムを指定します。次はその例です: `<item desc="home">{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}</item>`.
- `<template>` - この要素は、アプリケーションの初期化時に、指定されたテンプレートに基づきすべてのアイテムをキャッシュに読み込むよう指定します。次はその例です: `<template desc="layout">{3A45A723-64EE-4919-9D41-02FD40FD1466}</template>`.
- `<children>` - この要素は、指定されたアイテムのすべての子を読み込むように指定します。(次はその例です: `<children desc="main items">{110D559F-DEA5-42EA-9C1C-8A5DF7E70EF9}</children>`).
- `<childLimit>` - キャッシュに読み込む子の数を指定します。アイテムの子の数がこの値より多い場合、Sitecore は子をキャッシュしません。次はその例です: `<childLimit>100</childLimit>`.

5.3.3 プリフェッチ キャッシュの推奨プラクティス

Sitecore のプリフェッチ キャッシュの設定時のガイダンスとなる推奨プラクティスの一覧は次のとおりです:

- Sitecore のプリフェッチ キャッシングを活用するには、まず設定する必要があります。デフォルトのプリフェッチ キャッシング ファイルの情報は、多くの場合正確ではありません。たとえば、`/App_Config/Prefetch/Master.config` ファイルには `html` 関連コントロールの複数のエントリが含まれます。これらのコントロールは現在では Core データベースに含まれています。
- プリフェッチ キャッシュのサイズ設定時には、プリフェッチ キャッシュはアプリケーションの初期化時だけでなく、アプリケーションが実行されている間ずっと生成される場合があることを考慮する必要があります。
- アイテム キャッシュおよびデータ キャッシュと同様に、プリフェッチ キャッシュの拡大、サイズ、消去についても監視する必要があります。詳細は、上記の Sitecore キャッシュのチューニングを参照してください。
- Sitecore は、`<setting name="ContentEditor.RenderCollapsedSections" value="false" />` を `/web.config` ファイルに追加することを推奨しています。これは、デフォルトでは「true」に設定されている隠し設定です。この設定が「true」(デフォルト) のままの場合、コンテンツ エディターの実装により、Sitecore は壊れたアイテムの子をメモリー (キャッシュ) に読み込み続け、プリフェッチによりその子も読み込まれます。これにより、Sitecore は 2 つのレベルを余分に読み込むことになります。たとえば、ホーム アイテムを表示すると、動作が遅くなります。

5.4 出力 (レンダリング) キャッシュ設定のガイドライン

html 出力 (レンダリング) キャッシュのガイドラインの設定はタスクではなく、Sitecore 出力キャッシュについての入門、および設定、活用方法の推奨プラクティスの説明となります。

Sitecore 出力キャッシングについて十分に理解するには、*Sitecore* プレゼント コンポーネント リファレンスの第 4 章、出力キャッシングを読むことを推奨します。次の項を参照してください。

<http://sdn.sitecore.net/Reference/Sitecore%206/Presentation%20Component%20Reference.aspx>

5.4.1 出力 (レンダリング) キャッシュの理解

Sitecore 出力キャッシングを活用することにより、ウェブサイトのパフォーマンスを大幅に改善できます。レンダリング操作の結果を取得し、それらの結果をメモリーから表示すると、レンダリング コードを実行するよりもずっと早く処理できます。

Sitecore 出力キャッシングは、管理されている各ウェブサイトに関連付けられています。

デベロッパーは Sitecore 出力キャッシングを使用し、キャッシュするレンダリング コンポーネントをピックアップ、選択できます。あわせて、そのキャッシュの消去タイミングの規則を定義する VaryBy パラメーターを使用できます。これにより、同一の結果ウェブページ内のヘッダーやフッターなどの一般的なページ コンポーネントにキャッシュし、ニュース フィードなどの動的コンポーネントは動的に設定できます。

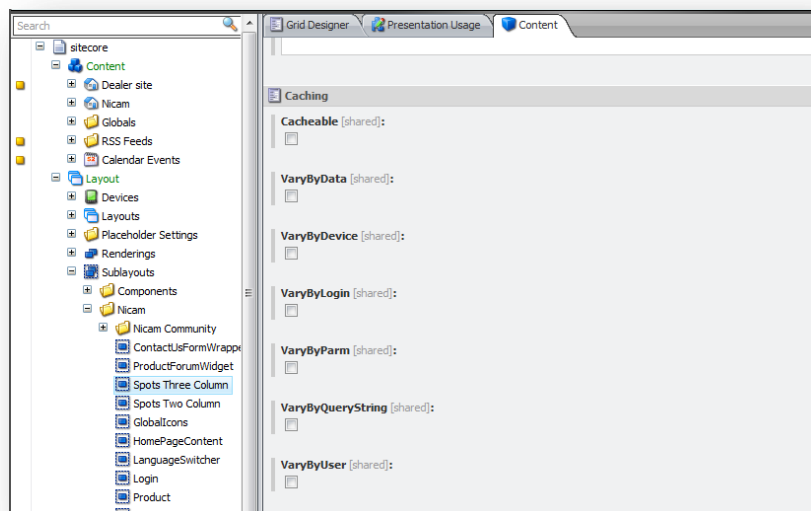
レイアウト エンジンのデフォルトの動作は、キャッシュ プレゼンテーション コンポーネントではありません。Sitecore 出力キャッシングを活用するには、プレゼンテーション コンポーネントを適切に設定する必要があります。この項では、設定方法のガイダンスと推奨プラクティスについて説明します。

Sitecore 出力キャッシングは、ASP.NET ページとフラグメントキャッシング (Web フォームと Web ユーザー コントロールにおいて、OutputCache デイレクティブと共に実装) と混同してはなりません。デベロッパーは、公開などの操作後に ASP.NET キャッシュを適切に消去する方法を理解せずに、Sitecore 出力キャッシングと ASP.NET ページとフラグメントキャッシングをあわせて使用してはなりません。

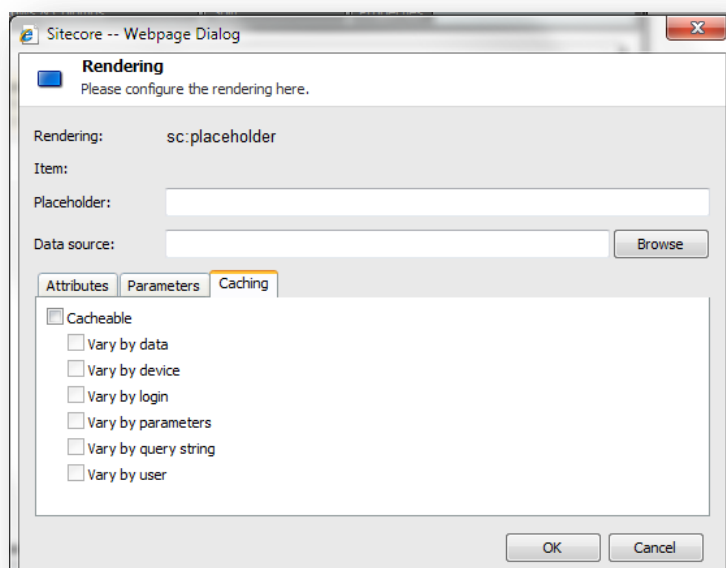
5.4.2 キャッシュ設定の設定場所および適用されるキャッシュ設定

デベロッパーは、3 つの場所で Sitecore の出力キャッシュを設定できます：

- サブレイアウトおよびレンダリング定義アイテムの **[キャッシング]** セクション。



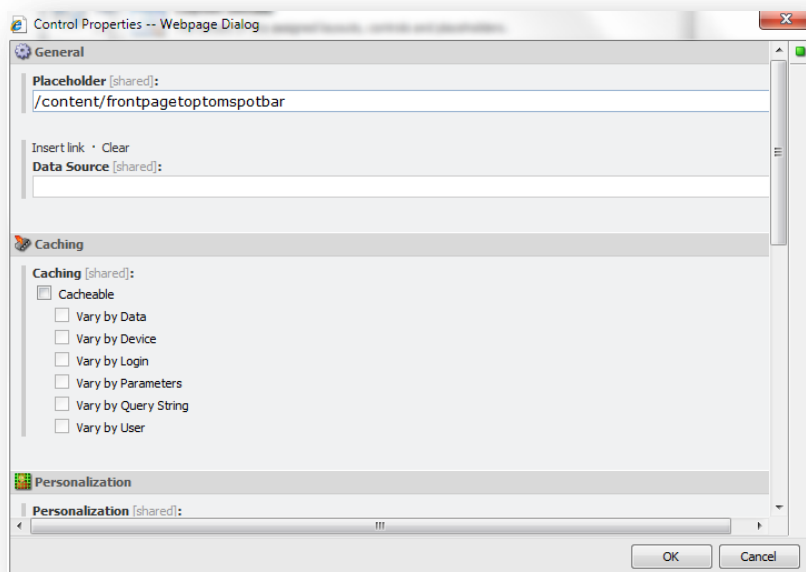
- プレゼンテーション コンポーネントをレイアウトまたはサブレイアウトに静的にバインドする際の、そのプロパティ。



- 宣言コード:

```
<sc:sublayoutrunat="server" renderingid="D366A65-54FF-49B5-A57F-2EBB9F426433%7d" placeholder="content" cacheable="true" varybydata="true" path="/layouts/Starter Kit/Sublayouts/Header Fixed.ascx" id="HeaderFixed"></sc:sublayout>
```

- プレゼンテーション コンポーネントを、レイアウト詳細のプレースホルダーにバインドする際の [コントロール プロパティ] ダイアログ ボックスの [キャッシング] セクション。



レイアウト エンジンは、次の 2 つの条件下で、定義アイテムの [キャッシング] セクションで定義されたキャッシュ基準を使用します:

- デベロッパーが、レンダリングをレイアウトまたはサブレイアウトに静的にバインドする場合、Sitecore はレンダリング定義アイテムからコントロールへキャッシング プロパティをコピーします (プレゼンテーション コンポーネントへの静的参照)。そのためには、定義アイテム内でキャッシング プロパティが設定されている必要があることにご注意ください。キャッシング プロパティの設定前にレンダリング定義アイテムでレイアウトまたはサブレイアウトが静的にバインドされている場合は、コピーされません。
- レイアウト詳細が、プレースホルダーに動的にバインドされているプレゼンテーション コンポーネントのキャッシング基準を指定しない場合。

レイアウト詳細を使用してプレースホルダーへのレンダリングを動的にバインドする場合、レイアウト詳細で明示的に定義されているキャッシュ設定が、レンダリング定義アイテムに定義されているキャッシュ設定を上書きします。定義アイテムに定義されているキャッシュ設定は、[コントロール プロパティ] ダイアログ ボックスの [キャッシング] セクションにキャッシングが設定されていない場合のみ適用されます。

5.4.3 出力キャッシング プロパティ

キャッシング可能なプレゼンテーション コンポーネントには次のプロパティがあります:

- **キャッシング可能** - **キャッシング可能**プロパティは、*VaryBy* プロパティの設定内容に関わらず、プレゼンテーション コンポーネントをキャッシュするかどうかを定義します。false に設定されている場合、要求ごとにプレゼンテーション コンポーネントを実行します。true に設定されている場合、最初の要求時にプレゼンテーション コンポーネントを実行し、その後の要求ではキャッシュから取得します。このプロパティが true に設定されていて、さらに 1 個以上の *VaryBy* プロパティが true に設定されている場合、プレゼンテーション コンポーネントを起動するか、それともキャッシュから取得するかを *VaryBy* プロパティが制御します。
- **VaryBy** プロパティ - **VaryBy** プロパティは、プレゼンテーション コンポーネントを実行するか、またはキャッシュから取得する場合を制御します。VaryBy プロパティは、キャッシングが可能が true に設定されている場合のみ有効になります。
 - **VaryByData** - 異なるデータソースで使用される場合、異なる出力結果を生成するコンポーネントに対して true に設定します。
 - **VaryByDevice** - 異なるデバイスで使用される場合、異なる出力結果を生成するコンポーネントに対して true に設定します。
 - **VaryByLogin** - 承認されたユーザーと未承認ユーザーで、異なる出力結果を生成するコンポーネントに対して true に設定します。レイアウト エンジンは、すべての匿名ユーザーを単一の承認済みユーザーとして処理することにご注意ください。
 - **VaryByParm** - 異なるレンダリング パラメーターが渡された場合、異なる出力結果を生成するコンポーネントに対して true に設定します。
 - **VaryByQueryString** - 異なるクエリ文字列パラメーターが渡された場合、異なる出力結果を生成するコンポーネントに対して true に設定します。
 - **VaryByUser** - 異なるユーザーに対して異なる出力結果を生成するコンポーネントに対して true に設定します。メモリーの過剰な使用を避けるために、*VaryByUser* は比較的ユーザー数の少ないソリューションでのみ使用してください。

メモ

Visual Studio を使用して、レイアウトまたはサブレイアウトに対してレンダリングを静的に配置している場合、キャッシング プロパティを手動で設定できます。次はその例です:<sc:sublayoutrunat="server" renderingid="D366A65-54FF-49B5-A57F-2EBB9F426433%7d" placeholder="content" cacheable="true" varybydata="true" path="/layouts/Starter Kit/Sublayouts/Header Fixed.ascx" id="HeaderFixed"></sc:sublayout>

5.4.4 出力キャッシングの推奨プラクティス

Sitecore の出力キャッシングの設定時のガイダンスとなる推奨プラクティスの一覧は次のとおりです:

- プレゼンテーション コンポーネントの起動に必要な CPU 使用率を削減することにより、Sitecore 出力キャッシングのレンダリング パフォーマンスに大きなメリットをもたらします。出力キャッシングの使用により最大の効果を得るには、プレゼンテーション コンポーネントをキャッシング可能にする前に適切なコーディング プラクティスに従っていることを確認する必要があります。
- htmlCacheSize プロパティが、すべてのレンダリングをキャッシングするのに十分な大きさであることを確認します。設定の影響の詳細については、「初期キャッシュ値の設定」のセクションを参照してください。
- 子コントロールを含むコンテナ (レイアウト、プレースホルダー、サブレイアウト) のキャッシングにより、以降の要求では、コンテナのすべてのレンダリング結果をキャッシュから取得することとなります。これには、コンテナ内のすべての子コントロールを含みます。
- stats.aspx ページを使用し、プレゼンテーション コンポーネントのキャッシング活動とレンダリング時間を監視します。
- カスタマイズされたキャッシング基準の作成の詳細は、次の Sitecore ブログを参照してください:<http://www.sitecore.net/Community/Technical-Blogs/John-West-Sitecore-Blog/Posts/2011/05/Custom-Caching-Criteria-with-the-Sitecore-ASPNET-CMS.aspx>

第 6 章

チューニング手順 - IIS 設定

チューニング手順 - IIS 設定には、IIS Web サーバーの設定をチェックするために設計された一連のタスクが含まれています。IIS を適切に設定することにより、Sitecore 実装は最善のパフォーマンスで実行できます。

この章には次のセクションがあります。

- HTTP キープアライブの有効化
- IIS の Expire Web コンテンツ ヘッダー
- IIS の静的なコンテンツの圧縮の有効化
- IIS の動的なコンテンツの圧縮の有効化 (オプション)

6.1 HTTP キープアライブの有効化

HTTP キープアライブを有効にすることにより、確立する必要のある接続数を削減できます。HTTP キープアライブが無効になっている場合、要求された Web ページ上のすべてのオブジェクトに対して新たな接続が確立されます。

6.1.1 必要なスキル

- 実行中の IIS マネージャー。

6.1.2 症状

- 要求ページの読み込み時間が常に長い。
- 低いパフォーマンス。

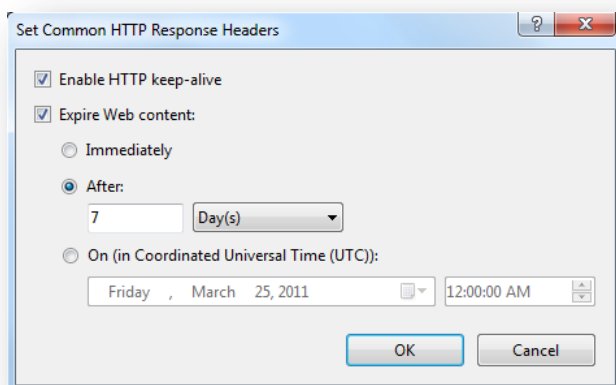
6.1.3 手順

HTTP キープアライブが有効になっているかどうかの確認方法:

1. **IIS マネージャー**を起動します。
2. **HTTP キープアライブ**が有効であるかを確認したいサイトに移動します。
3. **[HTTP 応答ヘッダー]** をダブルクリックします (IIS グループング内にあります)。
4. **[操作]** パネルで、**[共通ヘッダーの設定]** をクリックします。

6.1.4 結果の理解

[共通 HTTP 応答ヘッダーの設定] ダイアログが表示されます。次の例では、HTTP キープアライブが有効になっています。



6.1.5 推奨事項

Sitecore は、HTTP キープアライブを有効にすることを推奨しています。

6.1.6 解決方法

HTTP キープアライブを有効にする方法:

1. **IIS マネージャー**を起動します。
2. **HTTP キープアライブ**を有効に設定するサイトに移動します。
3. [**HTTP 応答ヘッダー**] をダブルクリックします (IIS グループ内にあります)。
4. [**操作**] パネルで、[**共通ヘッダーの設定**] をクリックします。
5. [**HTTP キープアライブの有効化**] を選択します。
6. [**OK**] をクリックします。

調査結果レポート

結果の記録

HTTP キープアライブが有効になっています ____ はい ____ いいえ

HTTP キープアライブが有効になっています = はい

OK。この結果は、推奨プラクティスに従い、HTTP キープアライブが有効であることを示しています。

HTTP キープアライブが有効になっています = いいえ

エラー。この結果は、HTTP キープアライブが有効になっていないことを示しています。Sitecore は、確立する必要がある接続数を削減するため、HTTP キープアライブを有効にすることを推奨しています。

6.2 IIS の Expire Web コンテンツ ヘッダー

Expire Web コンテンツ ヘッダー ([共通 HTTP 応答ヘッダー] 内にあります) では、Web ページ コンテンツの期限が切れた後に送信された要求に対し、要求 Web ページの新しいバージョンを返すかどうかを IIS がどのように決定するかを指定します。IIS はコンテンツの期限切れのための設定を使用し、送信前に各 Web ページをマークします。エンドユーザーのブラウザは期限切れマークを変換します。

Expire Web コンテンツを即時以外に設定することで、2 次アクセスの読み込み時間を 50% から 70% 削減できます。この設定は、動的に生成されたコンテンツには影響を与えません。

下記の手順は IIS 7.x 用であることにご注意ください。過去のバージョンで Expire Web コンテンツ ヘッダーを有効にする方法の詳細は、Microsoft のドキュメントを参照してください。

6.2.1 必要なスキル

- 実行中の IIS マネージャー。

6.2.2 症状

- 要求ページの読み込み時間が常に長い。
- 低いパフォーマンス。

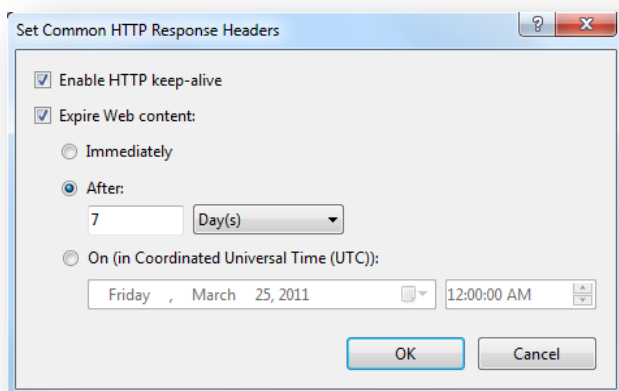
6.2.3 手順

Expire Web コンテンツ ヘッダーが有効になっているかどうかの確認方法:

1. **IIS マネージャー**を起動します。
2. **Expire Web コンテンツ ヘッダー**が有効であるかを確認したいサイトに移動します。
3. **[HTTP 応答ヘッダー]** をダブルクリックします (IIS グループ内にあります)。
4. [操作] パネルで、**[共通ヘッダーの設定]** をクリックします。

6.2.4 結果の理解

[共通 HTTP 応答ヘッダーの設定] ダイアログが表示されます。次の例では、Expire Web コンテンツが有効になっており、失効までの期間が 7 日に設定されています。



6.2.5 推奨事項

Sitecore は、Expire Web コンテンツの「失効までの期間」を 30 日に設定することを推奨しています。

6.2.6 解決方法

Expire Web コンテンツ ヘッダーを有効にする方法:

1. IIS マネージャーを起動します。
2. Expire Web コンテンツ ヘッダーを有効に設定するサイトに移動します。
3. [HTTP 応答ヘッダー] をダブルクリックします (IIS グループ内にあります)。
4. [操作] パネルで、[共通ヘッダーの設定] をクリックします。
5. [期限切れの Web コンテンツ] を選択します。
6. [失効までの期間] を選択します。
7. 日数を 30 に設定します。
8. [OK] をクリックします。

調査結果レポート

結果の記録

Expire Web コンテンツ ヘッダーが「失効までの期間」に設定されています ____ はい ____ いいえ

Expire Web コンテンツ ヘッダーが「失効までの期間」に設定されています = はい

OK。この結果は、推奨プラクティスに従い、Expire Web コンテンツ ヘッダーが有効になっており、「失効までの期間」に設定されていることを示しています。

Expire Web コンテンツ ヘッダーが「失効までの期間」に設定されています = いいえ

エラー。この結果は、Expire Web コンテンツ ヘッダーが有効になっていないか、「失効までの期間」に設定されていないことを示しています。Sitecore は、Expire Web コンテンツの「失効までの期間」を 30 日に設定することを推奨しています。

6.3 IIS の静的なコンテンツの圧縮の有効化

IIS の静的なコンテンツの圧縮を有効にすることにより、CPU リソースを低下させることなく、静的な応答を圧縮して、複数の要求にまたがりディスク内でキャッシュすることができます。

デフォルトでは、IIS の静的コンテンツの圧縮は有効に設定されています。このタスクでは、実際に有効になっているかどうかを確認します。

6.3.1 必要なスキル

- 実行中の IIS マネージャー。

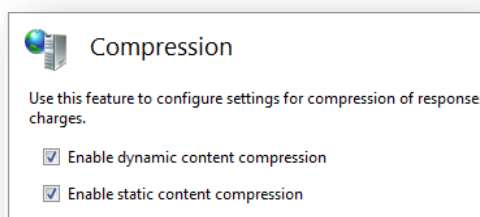
6.3.2 症状

- 要求ページの読み込み時間が常に長い。
- 低いパフォーマンス。

6.3.3 手順

静的なコンテンツの圧縮が有効になっているかどうかの確認方法:

1. **IIS マネージャー**を起動します。
2. **静的なコンテンツの圧縮**が有効であるかを確認したいサイトに移動します。
3. **[圧縮]** アイコンをダブルクリックします (IIS グループ内にあります)。
4. **[圧縮]** ダイアログが表示されます。次の例では、静的なコンテンツの圧縮が有効になっています。



6.3.4 推奨事項

Sitecore は、**IIS の静的なコンテンツの圧縮**を有効にすることを推奨しています。

6.3.5 解決方法

IIS の静的なコンテンツの圧縮を有効にする方法:

1. **IIS マネージャー**を起動します。
2. **静的なコンテンツの圧縮**が有効であるかを確認したいサイトに移動します。
3. **[圧縮]** アイコンをダブルクリックします (IIS グルーピング内にあります)。
4. **[静的なコンテンツの圧縮を有効にする]** チェックボックスをオンにします。
5. **[適用]** をクリックします。

調査結果レポート

結果の記録

IIS の静的なコンテンツの圧縮が有効になっています ____ はい ____ いいえ

IIS の静的なコンテンツの圧縮が有効になっています = はい

OK。この結果は、推奨プラクティスに従い、IIS の静的なコンテンツの圧縮が有効であることを示しています。

IIS の静的なコンテンツの圧縮が有効になっています = いいえ

エラー。この結果は、IIS の静的なコンテンツの圧縮が有効になっていないことを示しています。Sitecore は、IIS の静的なコンテンツの圧縮を有効にすることを推奨しています。

6.4 IIS の動的なコンテンツの圧縮の有効化 (オプション)

動的なコンテンツの圧縮の有効化は、CPU リソースがどの程度利用できるかによって効果が異なるため、オプションとなっています。

動的なコンテンツの圧縮は、動的コンテンツの応答を圧縮することにより、必要となる帯域幅を低減できます。動的コンテンツは要求/応答ごとに圧縮されるため、帯域幅を低減する代わりに CPU 利用率が高くなります。そのため、CPU 使用率が既に高いシステムの場合、動的なコンテンツの圧縮を有効にすべきではありません。

動的な圧縮を使用したメリットおよびその設定方法を紹介した 2 つの記事を、次の示します。

- <http://weblogs.asp.net/owscott/archive/2009/02/22/iis-7-compression-good-bad-how-much.aspx>
- <http://www.west-wind.com/weblog/posts/2011/May/05/Builtin-GZipDeflate-Compression-on-IIS-7x>

6.4.1 必要なスキル

- 実行中の IIS マネージャー。

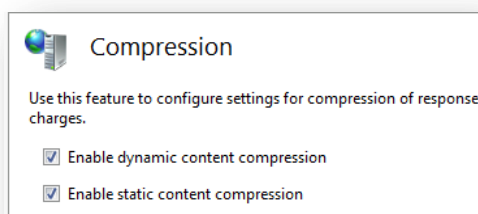
6.4.2 症状

- 要求ページの読み込み時間が常に長い。
- 低いパフォーマンス。

6.4.3 手順

動的なコンテンツの圧縮が有効になっているかどうかの確認方法:

1. **IIS マネージャー**を起動します。
2. **動的なコンテンツの圧縮**が有効であるかを確認したいサイトに移動します。
3. **[圧縮]** アイコンをダブルクリックします (IIS グルーピング内にあります)。
4. **[圧縮]** ダイアログが表示されます。次の例では、動的なコンテンツの圧縮が有効になっています。



6.4.4 推奨事項

Sitecore は、CPU 使用率が常に高い値を示しているシステム以外では **IIS の動的なコンテンツの圧縮**を有効にすることを推奨しています。

6.4.5 解決方法

IIS の静的なコンテンツの圧縮を有効にする方法:

1. **IIS マネージャー**を起動します。
2. **動的なコンテンツの圧縮**が有効であるかを確認したいサイトに移動します。
3. **[圧縮]** アイコンをダブルクリックします (IIS グループ内にあります)。
4. **[動的なコンテンツの圧縮を有効にする]** チェックボックスをオンにします。
5. **[動的なコンテンツの圧縮を有効にする]** チェックボックスがグレーアウトしている場合、**[コントロール パネル]**、**[プログラムの追加と削除]**、**[Windows コンポーネントの追加と削除]** で IIS の動的圧縮モジュールを追加してください。
6. **[適用]** をクリックします。

調査結果レポート

結果の記録

IIS の動的なコンテンツの圧縮が有効になっています ____ はい ____ いいえ

IIS の動的なコンテンツの圧縮が有効になっています = はい

OK。この結果は、推奨プラクティスに従い、IIS の動的なコンテンツの圧縮が有効であることを示しています。

IIS の動的なコンテンツの圧縮が有効になっています = いいえ

エラー。この結果は、IIS の動的なコンテンツの圧縮が有効になっていないことを示しています。Sitecore は、CPU 使用率が常に高い値を示しているシステム以外では IIS の動的なコンテンツの圧縮を有効にすることを推奨しています。

第7章

チューニング手順 - Sitecore のクライアント最適化

チューニング手順 - Sitecore のクライアント最適化には、Sitecore クライアントツールの設定と推奨プラクティスをチェックするために設計された一連のタスクが含まれています。

適切な設定と、次に示す推奨プラクティスにより、Sitecore クライアント ツール使用時の、ユーザーエクスペリエンスの最善なパフォーマンスを確保するのに役立ちます。

この章には次のセクションがあります。

- 実行時間の長いバリデーターのチェック
- 過剰なアイテム バージョンのチェック
- ノードごとの過剰アイテムのチェック
- その他のクライアント固有の最適化

7.1 実行時間の長いバリデーターのチェック

実行時間の長いバリデーターは、Sitecore コンテンツ エディターの使用時のパフォーマンスに悪影響を与えます。このタスクでは、そのような検証ルールを見つけるために設計されています。

7.1.1 必要なスキル

- Sitecore ログ解析の実務知識。

7.1.2 症状

- Sitecore コンテンツ エディター使用時の低いパフォーマンス。

7.1.3 実行時間の長い検証ルールのチェック手順

実行時間の長い検証ルールは、Sitecore ログ内で次のメッセージと共に表示されます: *Long running operation: Running Validation Rules*. このメッセージを見つけるには、お使いのテキスト エディターで上記の文字列を検索するか、Log Parser Lizard GUI でログ ファイルを解析します。

Log Parser Lizard GUI を使用してログ ファイルを解析する方法:

1. **Log Parser Lizard GUI** を起動します。
2. **新規クエリ**を作成します。
3. ログ入力フォーマットを **Multiline Regex (Log4Net)** 入力フォーマットに変更します。

次のスクリプトを実行します (#logs location# を "c:¥logs¥*.txt" などに変更してください):

```
SELECT *
FROM #Logs Location#
WHERE Message LIKE '%Long running operation: Running Validation Rules%'
```

7.1.4 結果の理解

出力結果は、次のようなものとなります:

Filename	RowNumber	ProcessID	Time	LogType	Message
C:\scinternet\sc\log.20110209.100803.txt	7,668	2976	13:56:19	WARN	Long running operation: Running Validation Rules
C:\scinternet\sc\log.20110209.100803.txt	10,576	2588	14:56:08	WARN	Long running operation: Running Validation Rules
C:\scinternet\sc\log.20110209.100803.txt	13,685	4020	15:36:42	WARN	Long running operation: Running Validation Rules
C:\scinternet\sc\log.20110210.150537.txt	330	1416	15:06:24	WARN	Long running operation: Running Validation Rules
C:\scinternet\sc\log.20110210.150537.txt	349	1416	15:06:27	WARN	Long running operation: Running Validation Rules

上記の表は、実行時間の長い検証ルールにより、5 つの警告が生じたことを示しています。どの検証ルールにより警告が生じたかは、ログ ファイルからは分かりません。

7.1.5 推奨事項

Sitecore は、Sitecore コンテンツ エディター使用時のパフォーマンス向上のため、実行時間の長い検証ルールを調査し、無効にすることを推奨しています。

7.1.6 解決方法

どの検証ルールから警告が生じたかを突き止めるには、多少の試行錯誤が必要となります。影響を与えている検証ルールを突き止めるのに役立つ方法を次に説明します：

1. カスタマイズされた検証ルールの実行時間が長くないかを確認します。長い場合は、無効にします。
2. 不要な標準のバリデーターは、アイテムの検証ルール セクションから削除して無効にします。
3. web.config ファイルで宣言されているバリデーターもあります (メディア ライブラリ検証ルールなど)。

調査結果レポート

結果の記録

ログ ファイルが、実行時間の長い検証ルールが存在を示しています __ はい __ いいえ

ログ ファイルが、実行時間の長い検証ルールが存在を示しています = いいえ

OK。Sitecore ソリューション内に、実行時間の長い検証ルールは存在しません。

ログ ファイルが、実行時間の長い検証ルールが存在を示しています = はい

エラー。ログ ファイルが、実行時間の長い検証ルールが存在を示しています。Sitecore は、警告を生じた検証ルールを調査し、無効にすることを推奨しています。

7.2 過剰なアイテム バージョンのチェック

1 つ以上のアイテムで過剰なアイテム バージョンを保持することは、パフォーマンスに悪影響を与える場合があります。特に、コンテンツ エディター内のコンテンツ ツリーの表示に関連するパフォーマンス、およびキャッシングに使用されるメモリ量に影響を与えます。

バージョン数が増えると、すべてのバージョンの処理に必要な時間も増えます。たとえば、コンテンツ エディターで表示または編集するアイテムを開く場合、表示されているバージョンだけではなく、そのアイテムに関連するすべてのバージョンも処理されます。

このタスクは、過剰なバージョン数、およびバージョン管理の共有ソース モジュールへのポインターを見つけ出すよう設計されています。

7.2.1 必要なスキル

- Sitecore コンテンツ エディターの実務知識。
- 共有ソース モジュールのインストールおよび設定経験。

7.2.2 症状

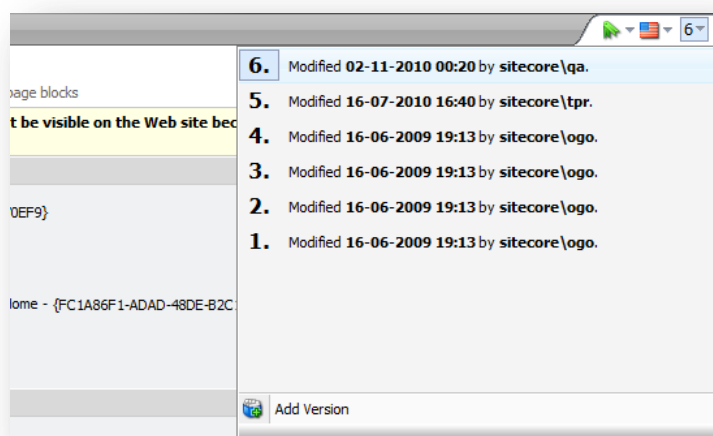
- Sitecore コンテンツ エディター使用時の低いパフォーマンス。

7.2.3 過剰なアイテム バージョンのチェック手順

通常、大幅に変更されたアイテムには過剰なバージョン数が存在します (サイトのホームページなど)。どのページが最も頻繁に更新されているかを、コンテンツ作成者に確認します。

1. **Sitecore コンテンツ エディター**を起動します。
2. コンテンツ ツリー内で、該当するアイテムに移動します。

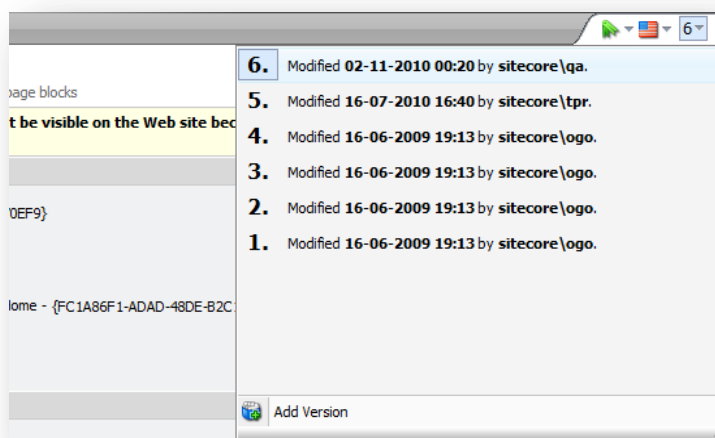
3. アイテムの右上隅に、表示されているアイテムのバージョン数を示すドロップダウン リストがあります：



4. アイテムのバージョン数にご注意ください (10 以上のバージョン数の場合は、過剰というフラグをつけます)。
5. このプロセスを、他の該当アイテムについても繰り返します。

7.2.4 結果の理解

出力結果は、次のようなものとなります：



この画面は、現在ホーム ページには 6 個のバージョンが保存されていることを示しています。

7.2.5 推奨事項

Sitecore は、特定のアイテムに対して 10 以下のバージョン数に限定することを推奨しています。ただし、会社のポリシーに保存すべきバージョン数が指定されている場合、推奨より多くのバージョンを保存することも可能です。また、Sitecore は保存バージョン数の管理のためにバージョン マネージャーの共有ソース モジュールの実行を推奨しています。

7.2.6 解決方法

保存するバージョン数を管理するには、次のバージョン マネージャー共有ソースを活用することができます:<http://trac.sitecore.net/VersionManager>

調査結果レポート

結果の記録

10 を越えるバージョンが保存されたアイテムが 1 つ以上あります __ はい __ いいえ

10 を越えるバージョンが保存されたアイテムが 1 つ以上あります = いいえ

OK。10 を越えるバージョンのあるアイテムはありません。

10 を越えるバージョンが保存されたアイテムが 1 つ以上あります = はい:

エラー。10 を越えるバージョンのあるアイテムが 1 個以上見つかりました。Sitecore は、すべてのアイテムのバージョン数を 10 個以下に抑え、保存バージョン数の管理のためにバージョン マネージャー共有ソース モジュールを使用することを推奨しています。

7.3 ノードごとの過剰アイテムのチェック

コンテンツ ツリーの構造は、いかなる親ノードの下にも過剰なアイテム数が配置されることがないよう、バランスよく配置しなければなりません。アイテム数は 100 を越えてはなりません。アイテムの整理にフォルダーを使用した構造を活用し、注意深く計画をたてる必要があります。

ある親ノード下に 100 を越える過剰なアイテム数がある場合、コンテンツ エディターでコンテンツ ツリー内を移動中の Sitecore ユーザーのパフォーマンスが低下します。

このタスクでは、コンテンツ ツリー内の過剰なアイテム数を見つけるために設計されています。

7.3.1 必要なスキル

- Sitecore コンテンツ エディターの実務知識。

7.3.2 症状

- Sitecore コンテンツ エディター使用時の低いパフォーマンス。

7.3.3 ノードごとの過剰なアイテム数のチェック手順

100 以上のアイテムを含むノードの検索は、手動タスクで実行することも、スクリプトを作成して自動処理することも可能です。過剰なアイテム数を含むノードを手動検索する方法：

1. Sitecore コンテンツ エディターを起動します。
2. コンテンツ ツリーで、すべてのノードを展開します。
3. 100 アイテム以上あるノードを探します。

スクリプトを使用して、子の数が過剰に多いノードを探すには、指定されたホーム ノードから、コンテンツ ツリーを繰り返し確認するスクリプトを作成します。次はその例です：

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Sitecore.Data.Items;
using Sitecore.Collections;

namespace Sitecore.Utilities
{
    public partial class NodeCount : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
```



```
{
    Item homeItem = Sitecore.Configuration.Factory.GetDatabase("master").Items["/sitecore/content/home"];
    getNumberOfChildren(homeItem);
}

private void getNumberOfChildren(Item node)
{
    ChildList children = new ChildList(node);
    foreach (Item child in children)
    {
        if (child.HasChildren && child.Children.Count >= 100)
        {
            Response.Write("Node: " + child.Name + " - " + child.Children.Count + "<br />");
            getNumberOfChildren(child);
        }
        else
        {
            Response.Write("<br />");
        }
    }
}
}
```

7.3.4 結果の理解

100 を越えるアイテム数を含むノードの数により、結果は異なります。100 を越えるアイテム数を含むすべてのノードには、注意のためにフラグを立てる必要があります。

7.3.5 推奨事項

Sitecore は、すべての親ノード下のアイテム数を 100 以下にすることを推奨しています。

調査結果レポート

結果の記録

100 を越えるアイテム数を含む親ノードが存在します __ はい __ いいえ

100 を越えるアイテム数を含む親ノードが存在します = いいえ

OK。100 を越えるアイテムを含む親ノードはありません。

100 を越えるアイテム数を含む親ノードが存在します = はい:

エラー。100 を越えるアイテムを含む親ノードが 1 個以上存在します。Sitecore は、すべての親ノード下のアイテム数を 100 個以下にすることを推奨しています。

7.4 その他のクライアント固有の最適化

次に、Sitecore クライアントのパフォーマンスを改善すると考えられる一連の最適化を示します。

このタスクでは、その設定を web.config ファイルに追加する方法について説明します。

7.4.1 必要なスキル

- Sitecore web.config ファイルの実務知識。

7.4.2 症状

- Sitecore コンテンツ エディター使用時の低いパフォーマンス。

7.4.3 ContentEditor.RenderCollapsedSections の設定

ContentEditor.RenderCollapsedSections は /web.config fileの隠し設定であり、デフォルトでは true に設定されています。<setting name="ContentEditor.RenderCollapsedSections" value="false" /> を /web.config ファイルに追加することで、特に大きなコンテンツ ツリーを持つクライアントのパフォーマンスを改善します。

また、値を false に設定することで、プリフェッチ キャッシングが指定された子をメモリーに 2 度追加してしまう状況を防ぐことができます。

コンテンツ エディターでスタンダード フィールドを表示してはなりません。

コンテンツ エディターのスタンダード フィールドの表示を無効にすることにより、アイテムごとの情報のレンダリングを減らすことが可能になり、パフォーマンスが改善します。

7.4.4 コンテンツ作成者は完全公開を使用不可

完全公開、または全アイテムの公開には、多くのリソースを消費します。完全公開中は、操作中にシステムを使用中のすべての Sitecore ユーザーのパフォーマンスが低下する恐れがあります。最適化公開または差分公開を使用することを強く推奨します。

7.4.5 ContentEditor.CheckHasChildrenOnTreeNodees の設定

この設定は、ツリー ノードのレンダリング時にエディターが HasChildren メソッドを使用するかどうかを指定します。この値を false に設定することでパフォーマンスが改善する場合があります。設定変更のメリットがあるかどうかを確認するための追加のテストが必要となります。