sitecore®
compelling web experiences™

Sitecore ECM 2.0

Performance Tuning Guide for ECM 2.0

A developer's guide to optimizing the performance of Sitecore ECM

The information contained in this document represents the current view of Sitecore Corporation on the issues discussed as of the date of publication and is subject to change at any time without notice. This document and its contents are provided AS IS without warranty of any kind, and should not be interpreted as an offer or commitment on the part of Sitecore, and Sitecore cannot guarantee the accuracy of any information presented. SITECORE MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

The descriptions of other companies' products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Sitecore. Sitecore cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

All trademarks are the property of their respective companies

©2012 Sitecore Corporation. All rights reserved.

Table of Contents

Chapter 1

Introduction

This guide is designed to help ensure the high performance of the Sitecore E-mail Campaign Manager (ECM) module. The goal of this document is to help you increase throughput from the Sitecore installation to the Mail Transfer Agent (MTA), not to show you how to increase the performance of the MTA.

This document is a fine tuning guide for the ECM and explains how to realize high-volume peak performance in your environment. Sitecore has a track record of increasing the performance for existing customers by 200%. This has been achieved by using this guide to adapt and optimize their configuration to match the local conditions, surrounding infrastructure, and environment of the ECM.

This guide contains the following chapters:

- **Chapter 1 — Introduction**
  This introduction to the document.

- **Chapter 2 — Potential Performance Gains**
  This chapter describes some areas where you can generally improve performance.

- **Chapter 3 — ECM Performance Management Tool**
  This chapter describes the ECM Performance Management Tool.

- **Chapter 4 — Tuning Procedures**
  This chapter describes how to tune the ECM to achieve peak performance while sending an email campaign to the MTA.

Chapter 2

Potential Performance Gains

This chapter describes some areas that can often decrease throughput between Sitecore ECM and the MTA and tells you how to solve these problems.

This chapter contains the following sections:

- Throughput Issues

2.1 Throughput Issues

When you are running an email campaign, the following issues can affect throughput:

- Fragmented indexes in the analytics database.
- Bandwidth limitations.
- CPU utilization.

2.1.1 Fragmented Indexes on the Analytics (DMS) Database

ECM reads and stores information about every recipient as an Automation State record in the Analytics database. It is therefore critical that you optimize the performance of the Digital Marketing System (DMS).

For more information about tuning the DMS, see the *Sitecore DMS Performance Tuning Guide*.

2.1.2 Bandwidth Limitations

Understanding the bandwidth, or bandwidth limitations that exist between the Sitecore implementation and the MTA can be the difference between a successful high output email campaign and an email campaign that crawls along.

Sitecore is capable of generating large quantities of high quality emails with attachments and/or embedded images at a high rate. For example, if the size of a single email is 100 KB and you are trying to achieve a rate of 40 emails per second, the bandwidth between the sending server and the receiving MTA must be able to handle 4 MB per sec. You must ensure that the available bandwidth can handle the load. Alternatively, you can decrease the size of the email or lower your expectations.

Solutions

Possible solutions to bandwidth limitations are:

- In low bandwidth situations, create text based emails that contain links to images.

  To create text based emails, navigate to the *Manager Root* item and clear the **Embed images** check box.

- Reduce the number of SMTP connections.

  This reduces the number of emails that are sent per second and means that it will take longer to complete a campaign. However, doing this will reduce the likelihood of sending errors occurring due to lack of bandwidth.

2.1.3 CPU Utilization

CPU utilization is a key performance metric that you can use to track CPU performance regressions or improvements, and is a useful benchmark for performance investigations and potential improvements when running the ECM code.

In general, there are two scenarios, which can indicate a potential performance improvement, namely, underutilization, and over utilization of the CPU. The former occurs when the average CPU utilization is low (below 50%), and the latter occurs when the average CPU utilization is high (above 90%). The following sections address these cases:

The target is to achieve 80% CPU utilization, which still allows the server to handle spikes.

Over Utilization of the CPU

CPU over utilization typically occurs when the load on the CPU(s) is greater than the capacity available. This is often seen on a Sitecore CMS server that is not only handling content creation, editing workflow, and publishing but is also used to create and send email campaigns. However, this is not limited to a shared server, since an improperly tuned dedicated ECM server can also experience over utilization of the CPU.

Solutions

- Throttle the CPU down by decreasing the number of `MaxGenerationThreads`.

   The `MaxGenerationThreads` setting determines how many threads the CPU processes simultaneously.

   For more information about decreasing the number of threads, see *Chapter 4 — Tuning Procedures.*

- Check to see if the email rendering times are slow — greater than 100ms. To check the email rendering times, view the rendering information on the `Sitecore stats.aspx` page, or by using Sitecore Rocks.

   Slow rendering times can cause excessive CPU usage.

   For more information about rendering times, see the section *Rendering Performance* in the *CMS Performance Tuning Guide.*

- To reduce the number of emails that are sent concurrently, decrease the number of SMTP connections.

- If your server is loaded with many other tasks besides sending emails, you should consider adding one or more dedicated servers to share the load of creating and sending campaigns.

   For more information about configuring servers, see the section *Configuring Dedicated Servers* in the *ECM Administrator's and Developer's Guide* for the version of the ECM that you are running.

Underutilization of the CPU

Underutilization of the CPU typically occurs when you are not taking advantage of the amount of available CPU resources to create and send email campaigns.

If you use a dedicated server to create and send email campaigns, CPU utilization will often be low — below 50%.

A properly tuned ECM solution allows you to gain peak performance when you create and send E-mail campaigns.

Solutions

- Throttle the CPU up by increasing the number of `MaxGenerationThreads`.

   The `MaxGenerationThreads` setting determines how many threads the CPU processes simultaneously.

   For more information about decreasing the number of threads, see *Chapter 4 — Tuning Procedures.*

- Increase the number of sending threads — `NumberThreads` setting.

Increase the number of SMTP connections — `SMTP.MaxConnections` setting.

Chapter 3

ECM Performance Management Tool

This chapter describes some new features in Sitecore CMS 6.6 that help you analyze the performance of your ECM implementation. The focus is on how to use and understand the new ECM Performance Management Tool to analyze performance.

This chapter contains the following sections:

- ECM Performance Measurement Tool

3.1 ECM Performance Measurement Tool

The ECM Performance Measurement Tool gives customers and consultants insight into the performance of their current email campaign and helps them evaluate their current configuration settings.

3.1.1 Understanding the ECM Performance Measurement Tool

This section describes how to launch the tool, the information it contains, and how this information relates to performance.

The information provided by the ECM Performance Measurement Tool should be used in conjunction with this tuning guide to insure that the ECM implementation is tuned for optimal performance.

Launching the ECM Performance Measurement Tool

To launch the ECM Performance Measurement Tool:

1. Open a web browser and navigate to:
   http://host_name/sitecore/admin/dispatchsummary.aspx.

   Replace *yoursite* with the name of the server that you use to dispatch email campaigns.

   Note: to open the tool in the German language, you must add the `?sc_lang=de-de` parameter to the end of this URL.

2. In a separate browser window, dispatch the email campaign that you wish to monitor.

   This can either be to an MTA, or through the MTA emulator.

   For more information about dispatching an email campaign, see *Chapter 4 — Tuning Procedures*.

3. To see the current performance statistics in the ECM Performance Measurement Tool, refresh the page while the email campaign is being dispatched.

The ECM Performance Measurement Tool contains the following fields:

Field	Description	Sample Value
CPU Cores	The number of CPU cores that are available on the server that is dispatching the email campaign.	4

Field	Description	Sample Value
Automation State Bulk Fetch	The number of queued recipients to fetch from the *Analytics* database in a single request. This value corresponds to the `RecipentRequestSize` setting in the `Sitecore.EmailCampaign.config` file.	1000
CPU Usage	The CPU usage — in %. This value is the average for the past 5 seconds.	91% on this server
Number of Threads	The maximum number of threads that you can use for sending messages. This value corresponds to the `NumberThreads` setting in the `Sitecore.EmailCampaign.config` file.	50
Max Generating Threads	The number of threads that can generate messages at the same time. This value corresponds to the `MaxGenerationThreads` setting in the `Sitecore.EmailCampaign.config` file.	10
Max SMTP Connections	The maximum number of concurrent SMTP connections that can be opened. This value corresponds to the `SMTP.MaxConnections` setting in the `Sitecore.EmailCampaign.config` file.	100
Send Emulation Time	The min and max send times. This information is taken from the `MtaEmulation.MinSendTime` and `MtaEmulation.MaxSendTime` settings in the `Sitecore.EmailCampaign.config` file. The first value, [0...2] represents the minimum amount of time, in ms, that is takes to emulate sending a single message. The second value [0…2] represents the maximum amount of time, in ms, that it takes to emulate sending a single message.	[0…2] ms
Message	The name of the dispatch that is currently running.	Newsletter Message
Start Time	The start date and time of the dispatch that is currently running.	2012-02-02 17:57:20
Duration	The length of time that the current dispatch has been running for.	00:00:32.9347490
Fetch Automation States - Total	The total time (Lock, Process) it takes to retrieve the ID of an automation state.	0 ms
Fetch Automation States - Lock	The time it takes to acquire the lock that you need to retrieve the ID of an automation state.	0 ms
Fetch Automation States - Process	The time it takes to process the retrieval of an automation state ID.	0 ms

Field	Description	Sample Value
Fetching Speed	The number of automation state IDs retrieved per second.	145112 state/s
Sending	The number of emails sent per second by this server. This value is the overall performance indicator for the current settings.	45 emails/s on this server
Total mails sent	The total number of emails sent in the current dispatch.	1441 emails on this server
Fetch Recipient - Wait	The amount of time spent in a wait state while retrieving a single recipient.	N/A
Fetch Recipient - Process	The time used to retrieve a single recipient.	3 ms
Fetching	The rate at which recipients are fetched per second. This value helps you see if there is enough throughput to the other parts of the application. If the value is low, it means there could be a database performance issue.	43 recipient/s
Generate Single Email (avg) - Total	The average total time it takes to generate a single email message. This is the sum of the time that you must wait for the thread plus the time to generate the email.	225 ms
Generate Single Email (avg) - Wait	The average time you must wait to launch the thread that generates a single email message.	0 ms
Generate Single Email (avg) - Process	The average time it takes to generate an email message. This time is the sum of time it takes to load the recipient, get the page, render the page, collect the attached files (if any), generate the MIME section plus some other operations. In the current example, Load the recipient = 3 ms Get the page= 180 ms Collect the file = 3 ms Generate the MIME section = 34 ms Total = 220 The remaining 5 ms are the other operations involved in the generating an email.	225 ms
Generate Single Email (avg) – Process – Load User	The average time required to load the recipient information while generating a single email message.	3 ms
Generate Single Email (avg) – Process – GetPage	The average time required to render the page, based on recipient data, while generating a single email message.	180 ms

Field	Description	Sample Value
Generate Single Email (avg) – Process – Collect files	The average time required to collect the files while generating a single email message.	3 ms
Generate Single Email (avg) – Process – Generate MIME	The average time required to create the MIME section while generating a single email message.	34 ms
Generating	The number of emails generated per second. This value helps you see if there is enough throughput in the sending part of ECM. If the speed is lower than expected, it could mean that the sending speed is affected.	43 email content/s
Send Email – Failed send attempts	The total number of send attempts that failed. This value helps you to identify any network or SMTP server problems.	0
Send Email – Wait	The time you must wait to acquire access to the connection pool. This is the time spent waiting before any send steps are performed.	0 ms
Send Email – Process	The time spent sending the email.	21 ms
Sending	The number of emails generated per second. The difference between this value and the overall send rate for the server is that this value does not take into account the time required to reset the threads and perform any other operations before a new thread is ready to start a new email.	43 email/s
Required Bandwidth	The bandwidth required to send emails at the indicated send speed. If the value is equal to or higher than the available network bandwidth and you want to increase the send speed, you must increase the network bandwidth.	1004 KB/s

Chapter 4

Tuning Procedures

This chapter describes how to tune the ECM to achieve peak performance while sending an email campaign to the MTA.

This chapter includes the following sections:

- Checking the ECM Version

- DMS Performance Tuning

- Tuning the ECM

- MTA Emulation

- Tuning a Shared Environment

- Tuning a Dedicated Environment

- Common Performance Improvements

4.1 Checking the ECM Version

Sitecore recommends that you use the latest version of the E-mail Campaign Manager module. Using the latest version ensures that you have all the newest stability and performance improvements.

4.1.1 Procedure to Check ECM Version

To see which version of the ECM module you are using:

1. Navigate to the bin directory of the website. This is typically `/Web Root/bin`.

2. Right-click the `Sitecore.E-mailCampaign.dll` file and then click **Properties**.

3. In the **Properties** window, click the **Details** tab.

   The version information is listed here.

4.1.2 Sitecore Recommendation

Sitecore recommends that you run the latest version of the ECM module that is associated with the Sitecore CMS version that you are running. To find the latest version, go to: http://sdn.sitecore.net/Products/ECM.aspx

4.1.3 Report Findings

Record the results:

The ECM product version = _____

The ECM product version is the latest available:

OK. The results show that the latest ECM product version is installed.

The ECM product version is not the latest available:

Error. The results show that the latest ECM product version is not installed. Sitecore recommends that you use the latest ECM product version to benefit from the latest stability and performance improvements.

4.2 DMS Performance Tuning

ECM works closely with the *Analytics* database. Performance issues related to the *Analytics* database can quickly create performance issues for the ECM. It is important that you tune your DMS implementation.

You must tune your DMS implementation before you tune the ECM module, otherwise the ECM tuning effort may generate false results.

For information on how to tune your DMS implementation, see to the *DMS Performance Tuning Guide.*

4.2.1 Report Findings

Record the results:

The DMS has been tuned in accordance to the DMS Performance Tuning Guide __Yes __No

The DMS has been tuned in accordance to the DMS Performance Tuning Guide = Yes

OK. The DMS has been properly tuned.

DMS has been tuned in accordance to the DMS Performance Tuning Guide = No

Error. The DMS has not been tuned according to the DMS Performance Tuning Guide. Sitecore recommends that you tune the DMS before you tune the ECM module.

4.3 Tuning the ECM

This section describes how to tune your ECM implementation to achieve the maximum possible throughput, given the type of environment that you are running in.

ECM environments are either shared or dedicated.

Shared Environment

In a shared environment, where you use the same server to create and send email campaigns and perform other CMS activities — for example, creating content, editing content, workflow, publishing, delivery, and so on, CPU utilization must remain at a percentage that is low enough to allow all these different activities to take place.

ECM — Targeted CPU Utilization: ~30-40% while dispatching emails.

Dedicated Environment

In a dedicated environment that is only used to create and send email campaigns, CPU utilization must be maximized — but not over utilized — to ensure the highest possible output.

ECM — Targeted CPU Utilization: ~80% while dispatching emails.

Tuning is an iterative process that starts with you setting the initial configuration values, depending on whether you are running in a shared or dedicated environment.

The initial settings in the ECM configuration file are based on the default settings shipped with ECM. These settings are based on Sitecore's internal test and case studies for enterprise customers, and have been optimized for a dedicated environment.

Round-trip

In the tuning example in this document, we assume that the round trip time between the ECM implementation and the MTA is 1 second. We also tell you how to calculate the target rate by pinging the SAC MTA. The actual sending rates may be higher or lower depending on several factors, including but not limited to:

- Bandwidth limitations.
- Round-trip time.
- The size and complexity of the emails.
- The time it takes to render emails and the cache the rendered output.
- System size and the available resources.
- Network speed
- Analytics database maintenance.

4.3.1 Prerequisites

Round-trip

You must know the round trip time it takes to send an email to the MTA and get a response back.

This is important in determining the number of sending threads that may be required to achieve the performance goals you are aiming for.

Bandwidth Limitations

You must know your bandwidth limitations. If you are trying to send 32 Mbps (4MBytes/sec) worth of emails, but only have enough bandwidth to handle 8 Mbps, you will not reach your goals.

In cases where you only have limited bandwidth available, you should consider creating a text based email with links instead of creating a rich email with imbedded images. To create a text based email, in the **Manager Root** item, clear the **Embed images** check box.

CPU Utilization

You must know the CPU utilization during the periods when you are sending emails.

This will differ for a shared environment from a dedicated environment. In a shared environment, the optimal condition would be to schedule sending the email campaign during periods when no other CMS activities are taking place.

E-mail Campaign Manager

You must know how to use the E-mail Campaign Manager.

This document does not describe how to create campaigns and emails, or how to schedule and run them. For information about using the E-mail Campaign Manager, see the documentation for the version you are using.

4.4 MTA Emulation

The ECM module lets you test campaign throughput by emulating an MTA. MTA emulation allows you to mimic the round trip time required to send an email from Sitecore CMS to an MTA.

There are two ways to configure emulation:

- Single dispatch.
- Server level dispatch.

This section describes how to configure the ECM module to run in MTA emulation mode.
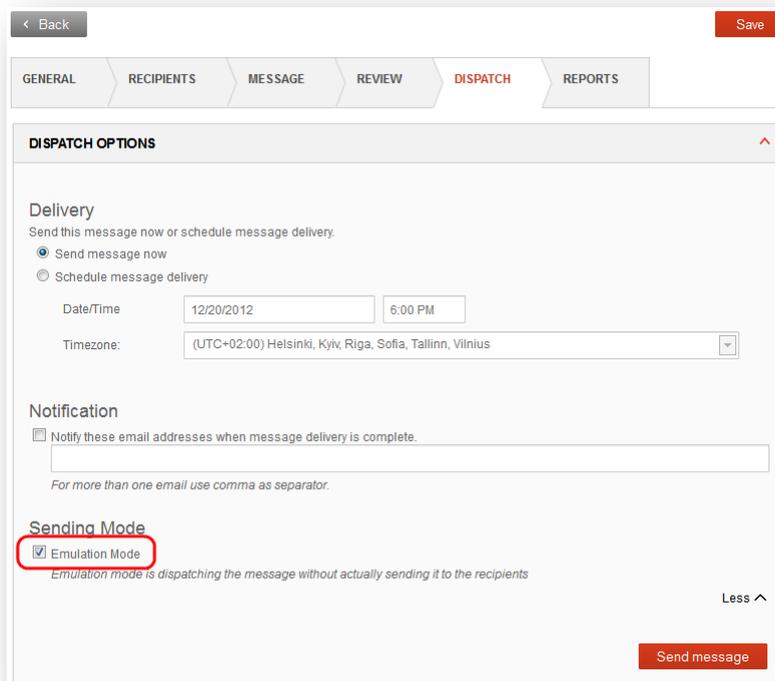
Prerequisite

To run in MTA emulation mode, the ECM module must be able to connect to an MTA. This can be a local MTA, a remote MTA, or a Sitecore Application Center (SAC) account.

If a connection is not available, a *Failed to connect to SMTP Server* error will occur.

Enabling MTA Emulation for a Single Dispatch

To enable MTA Emulation Mode for a single dispatch:

1. Select a message.
2. In the **Dispatch** tab, click **More**. The ECM displays the **Sending Mode** section.
3. Select the **Emulation Mode** checkbox to enable MTA Emulation Mode.

Enabling MTA Emulation for the Server Level Dispatch

To enable the MTA Emulation Mode for a server level dispatch:

1. Open up the `Sitecore.EmailCampaign.config` file.

This file is stored in the `<web root>/App_Config/Includes` directory.

2. Change the `MtaEmulation.Active` setting to `true`.

3. Save your changes.

MTA Emulation Mode Configuration

The MTA configuration settings are in the `Sitecore.EmailCampaign.config` file:

```
<!--The minimum amount of time to emulate a single sending (milliseconds).-->
<setting name="MtaEmulation.MinSendTime" value="200" />

<!--The maximum amount of time to emulate a single sending (milliseconds).-->
<setting name="MtaEmulation.MaxSendTime" value="400" />

<!--The probability of a connection fail (%).-->
<setting name="MtaEmulation.FailProbability" value="0.01" />
```

Setting	Value
MtaEmulation.MinSendTime	The minimum amount of time, in milliseconds, it takes to emulate sending a single email. Use this in conjunction with the `MtaEmulation.MaxSendTime` setting to mimic the Sitecore CMS to MTA round trip time.
MtaEmulation.MaxSendTime	The maximum amount of time, in milliseconds, it takes to emulate sending a single email.
MtaEmulation.FailProbability	Allows you to introduce the probability of connection failure into your emulation.

4.4.1 Determining MinSendTime and MaxSendTime when using SAC

You can measure round-trip time (RTT) by starting a timer when you transmit a packet to the server and stopping the timer when you receive the answer. You can use a ping command to measure this locally on the server running ECM. The RTT contains the time to transmit a packet, process it on the server running ECM, and receive the answer from the Sitecore Application Center (SAC) server.

Note
If you are using a different MTA than SAC, you must change the ping address.

You can set the `MinSendTime` and `MaxSendTime` for MTA emulation by pinging the SAC server:

1. Open the **Command** window.

2. Use a larger package size to send a larger package to the SCA.

   At the prompt enter: `ping –l 100 –n 50 core-01-ewr.dyndns.com`

   Write down your min. time: min1 = _____

   Write down your max time: max1 = _____

   Write down package size: psize1 = _____ bytes

3. Use a larger package size to send a larger package to the SCA.

   At the prompt enter: `ping –l 1400 –n 50 core-01-ewr.dyndns.com`

   Write down your min. time: min2 = _____

   Write down your max time: max2 = _____

   Write down package size: psize2 = _____ bytes

Note
To verify your results, make a sanity check against your upload connection speed (the bandwidth is likely to be much higher than observed) and run an online speed test on the Internet. A speed test is probably available at your ISP's web-site. The bandwidth is likely to be slightly higher than observed.

4. Use the following formula to calculate the bandwidth for min. and max.:

```
bandwidth min = 2 * (psize1 – psize2) / (min1 – min2) Bytes/sec
```

```
bandwidth max = 2 * (psize1 – psize2) / (max1 – max2) Bytes/sec
```

5. Estimate the size of the messages you are sending.

For example, ECM message size = 12,800 bytes.

ECM message size = _____ bytes

6. Estimate the `MinSendTime` and `MaxSendTime` values.

The minimum amount of time it takes to emulate sending a single message (in milliseconds):

```
MtaEmulation MinSendTime = ECM message size [bytes] / bandwidth min
[bytes/sec] / 1000 [sec/ms]
```

The maximum amount of time it takes to emulate sending a single message (in milliseconds):

```
MtaEmulation MaxSendTime = ECM message size [bytes] / bandwidth max
[bytes/sec] / 1000 [sec/ms]
```

4.4.2   Configuration Settings

The configuration settings are in the `Sitecore.EmailCampaign.config` file.

The settings that focus on tuning are:

```xml
<!--The maximum number of concurrent SMTP connections allowed to open-->
<setting name="SMTP.MaxConnections" value="100" />

<!--The number of threads that you can use for sending messages-->
<setting name="NumberThreads" value="100" />

<!--Specifies how many sending threads can generate messages at the same time.
        Default value: Environment.ProcessorCount * 2-->
<setting name="MaxGenerationThreads" value="2" />

<!--The number of queued recipients to get from the Analytics database via a single
request.-->
<setting name="RecipientsRequestSize" value="500" />

<!--The amount of time to wait between sending messages (ms)-->
<setting name="Sleep" value="0" />
```

Setting	Description
SMTP.MaxConnections	The maximum number of concurrent SMTP connections allowed.
NumberThreads	The total number of sending threads used for sending messages.
MaxGenerationThreads	The CPU throttle — specifies how many sending threads can generate messages at the same time.
RecipientRequestSize	The number of recipients that are queued from the analytics database in a single request.
Sleep	The amount of time, in milliseconds, to wait between sending messages.

4.5 Tuning a Shared Environment

Tuning a shared environment, or server, involves making sure that there are enough CPU resources available to not only run an email campaign, but also allow you to run the other CMS related tasks. If your CMS server has reached the limits of CPU utilization, you should consider running a dedicated ECM server to create and send E-mail campaigns.

4.5.1 Initial Configuration Values for Tuning

For demonstration purposes, the initial configuration values are based on a sending rate of 40 emails per second.

Setting	Initial Value
SMTP.MaxConnections	Same as `NumberThreads`
NumberThreads	100
MaxGenerationThreads	Equal to the number of available CPU cores. **Note** By default, the `MaxGenerationThreads` setting is commented out and has a default value of `Environment.ProcessorCount * 2`. The number of available CPU cores will approximate to 50% CPU utilization, thereby allowing you to run other CMS in the same environment. Depending on the CPU architecture you might need 2*CPU cores to reach 50% CPU utilization. If you are in doubt about your server's processor capacity, refer to the vendor's web-page or use the following code to create a console application and evaluate the capacity. ```using System;

namespace Sitecore.Environment.ProcessorCount
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("The number of processors on the current machine is: ");

Console.WriteLine(System.Environment.ProcessorCount);
            Console.ReadKey();
        }
    }
}``` |
| RecipientRequestSize | 500 |
| Sleep | 0 |

4.5.2 Testing

- Enable and configure MTA emulation mode to mimic the round trip characteristics of your Sitecore CMS to MTA connection.

  For information on how to enable and configure MTA emulation mode, see the section *MTA Emulation*.

- Create an email that is the same size and uses the same rendering complexity as the email messages that you typically send. You can also use an existing email message.

  You should also create a group of recipients to dispatch the messages to. The group of recipients should be large enough to allow for 2 or 3 minutes of run time. No email messages are sent in emulation mode.

- Before you dispatch the messages, on the computer that is creating and sending the emails, open the **Task Manager** and click the **Performance** tab.

- Dispatch the messages and monitor the CPU utilization.

- Review the next section *Understanding the Results* to find out where modifications need to be made, and then run the test again.

4.5.3 Understanding the Results

To understand the results of the test, you must study two pieces of information:

- The sending rate.
- CPU utilization during the email campaign.

Calculating the Send Rate

You can find the start and finish times of the email campaign in the Sitecore log file:

1. Open the Sitecore log file that is relevant for the test that you just performed.

2. Search the log file for the term *EmailCampaign: Dispatch Message*.

   There should be two entries that relate to the test. Make sure that the time stamp correlates to when the test was run.

3. The first entry will show when the email campaign started.

   For example:

   ```
   ManagedPoolThread #1 10:26:22 INFO  EmailCampaign: Dispatch Message
   (NewsHotfix_semch3): Started
   ```

4. The second entry will show when the email campaign finished.

   To calculate the send rate: `number of recipients / (finish time in seconds – start time in seconds)`

   For example:

   ```
   ManagedPoolThread #1 10:29:08 INFO  EmailCampaign: Dispatch Message
   (NewsHotfix_semch3): Finished
   ```

CPU Utilization

In this example, the sending rate is too low.

To increase the number of sending threads:

- Make sure that the MTA emulation mode is setup correctly to emulate the minimum and maximum send times.

For information on how to enable and configure MTA emulation mode, see the section *MTA Emulation*.

- Increase the number of `MaxGenerationThreads`.

    This increases the number of concurrent threads that are processed.

- Increase the `NumberThreads` configuration value.

    Ensure that the `SMTP.MaxConnections` value is greater than or equal to the `NumberThreads`.

    Ensure that `NumberThreads` is larger than `MaxGenerationThreads`.

- Rerun the test.

CPU Utilization is High

High CPU utilization, 90-95%, on a shared environment could mean that the amount of CPU resources used to send an email campaign can limit your ability to perform any other CMS activities. A targeted goal of 80% CPU utilization will allow the server to handle spikes in the load.

If CPU utilization is above 90%, you must determine where the bottleneck is, and rectify the problem.

Here is a list of possible solutions. You should view them in this order.

- **MaxGenerationThreads**

    Decrease the `MaxGenerationThreads` value. This reduces the number of concurrent threads that are processed.

    In a shared environment, reduce the `MaxGenerationThreads` to the number of available CPU cores or by half.

    Rerun the test after making the configuration changes.

CPU Utilization is Low

Underutilization of the CPU typically occurs when you are not taking advantage of the available CPU resources when you create and send email campaigns.

Possible solutions to increasing the CPU utilization include:

- **MaxGenerationThreads**

    Increase the `MaxGenerationThreads` value. This increases the number of concurrent threads that are processed.

    For optimal performance, the maximum value of `MaxGenerationThreads` should be increased to is: *2 * number of available CPU cores*.

- **NumberThreads**

    Increase the `NumberThreads` value.

    This increases the number of available sending threads and raises your sending rate to a higher level.

    You must also take bandwidth considerations into account so that you do not flood the available bandwidth between the ECM implementation and the MTA.

- **SMTP.MaxConnection**s

    Increase the number of SMTP connections so that the number of connections is equal to or greater than the `NumberThreads` setting.

4.6 Tuning a Dedicated Environment

Tuning a dedicated environment, or server, involves making sure that the available CPU resources are utilized to achieve the highest degree of performance. If the targeted sending rate is too high, additional dedicated servers may be required.

4.6.1 Initial Configuration Values for Tuning

For demonstration purposes, the initial configuration values are based on a sending rate of 40 emails per second.

Setting	Initial Value
SMTP.MaxConnections	Same as NumberThreads
NumberThreads	100

MaxGenerationThreads	Equal to the *2 * number of available CPU cores.*
	
	A value of *2 * number of available CPU cores* will approximate 100% CPU utilization and give optimal performance.
	
	Depending on the CPU architecture, a value of *4 * number of available CPU cores* might be needed to reach full CPU utilization.
	
	If you are in doubt about processor capacity of your server, please refer to the vendors web-page or use the following code to create a console application and evaluate the capacity.
	
	`using System;`
	
	`namespace Sitecore.Environment.ProcessorCount`
	`{`
	`    class Program`
	`    {`
	`        static void Main(string[] args)`
	`        {`
	`            Console.WriteLine("The number of processors on the current machine is: ");`
	
	`Console.WriteLine(System.Environment.ProcessorCount);`
	`            Console.ReadKey();`
	`        }`
	`    }`
	`}`

RecipientRequestSize	500
Sleep	0
Setting	Initial Value

Testing

- Enable and configure the MTA emulation mode to mimic the round trip characteristics of your Sitecore CMS to MTA connection.

For information on how to enable and configure MTA emulation mode, see the section *MTA Emulation*.

- Create an email that is the same size and uses the same rendering complexity as the email messages that you typically send. You can also use an existing email message.

  You should also create a group of recipients to dispatch the messages to. The group of recipients should be large enough to allow for 2 or 3 minutes of run time. No email messages are sent when you run in emulation mode.

- Before you dispatch the messages, on the computer that is creating and sending the emails, open the **Task Manager** and click the **Performance** tab.

- Dispatch the messages and monitor the CPU utilization.

- Review the next section *Understanding the Results* to find out where modifications need to be made, and then run the test again.

Understanding the Results

To understand the results of the test, you must look at two pieces of information:

- The send rate.
- CPU utilization during the email campaign.

Calculating the Send Rate

You can find the start and finish times of the email campaign in the Sitecore log file:

1. Open the Sitecore log file that is relevant for the test that you just performed.
2. Search the log file for the term *EmailCampaign: Dispatch Message*.

   There should be two entries that relate to the test. Make sure that the time stamp correlates to when the test was run.

3. The first entry will show when the email campaign started.

   For example:

   ```
   ManagedPoolThread #1 10:26:22 INFO  EmailCampaign: Dispatch Message
   (NewsHotfix_semch3): Started
   ```

4. The second entry will show when the email campaign finished.

   To calculate the send rate: `number of recipients / (finish time in seconds – start time in seconds)`

   For example:

   ```
   ManagedPoolThread #1 10:29:08 INFO  EmailCampaign: Dispatch Message
   (NewsHotfix_semch3): Finished
   ```

CPU Utilization is Low

Underutilization of the CPU typically occurs when you are not taking advantage of the available CPU resources when you create and send email campaigns.

Possible solutions to increasing the CPU utilization include:

- Make sure that the MTA emulation mode is setup correctly to emulate the minimum and maximum send times.

  For information on how to enable and configure MTA emulation mode, see the section *MTA Emulation*.

- **MaxGenerationThreads**

Increase the `MaxGenerationThreads` value. This increases the number of concurrent threads that are processed.

If `MaxGenerationThreads` is set too high, performance degradation may occur as a result of extensive thread context switching.

- **NumberThreads**

  Increase the `NumberThreads` value.

  This increases the number of available sending threads and raises your target sending rate to a higher level.

  Make sure that the `SMTP.MaxConnections` is greater than or equal to the `NumberThreads`, and that `NumberThreads` is larger than `MaxGenerationThreads`.

- Rerun the test.

CPU Utilization is High

High CPU utilization on a dedicated server could mean that the server cannot handle a spike in load. A targeted goal of 80% CPU utilization will allow the server to handle spikes in the load.

If CPU utilization is above 90-95%, you should throttle down CPU usage.

Possible causes and solutions that should be viewed in order.

- **MaxGenerationThreads**

  Decrease the `MaxGenerationThreads` value. This reduces the number of concurrent threads that are processed.

  In a dedicated environment, reduce the `MaxGenerationThreads` to somewhere between 2 * and 4 * the number of available CPU cores.

  Rerun the test after making the configuration changes.

CPU Utilization is Low

Underutilization of the CPU typically occurs when you are not taking advantage of the available CPU resources when you create and send email campaigns.

A properly tuned ECM solution allows you to gain peak performance when creating and sending email campaigns.

Possible solutions to increasing the CPU utilization include:

- **MaxGenerationThreads**

  Increase the `MaxGenerationThreads` value. This increases the number of concurrent threads that are processed.

  For optimal performance, the maximum value of `MaxGenerationThreads` should be increased to is: *2 * number of available CPU cores*.

- **NumberThreads**

  Increase the `NumberThreads` value.

  This increases the number of available sending threads and raises your sending rate to a higher level.

  You must also take bandwidth considerations into account so that you do not flood the available bandwidth between the ECM implementation and the MTA.

4.7 Common Performance Improvements

Normally, the bottleneck in a server that sends emails is the available CPU resources. In both shared and dedicated environments, you can increase the number of emails the CPU can generate per second by using the same Sitecore techniques that you use to optimize a web-page:

- **Rendering**

  Rendering an email is the same as rendering a web page. Rendering can demand a lot of CPU resources particularly if you did not think about performance when you created the page or email.

  To see whether or not rendering an email demands excessive CPU usage, create a very simple, small text based version of the email and rerun the test.

  If CPU utilization drops, then there is an issue with the rendering of the emails.

  For more information about testing and creating renderings, see the *CMS Performance Tuning Guide* and the article *Recommend Practices*.

To improve performance on both shared and dedicated environments you can add more shared or dedicated servers.

- **Dedicated Server(s)**

  If you have removed all the bottlenecks described in this manual and lowered your CPU utilization but still not achieved the send rate you wanted, you should consider adding more dedicated servers.

  For information on how to setup and configure a dedicated server, see:

  o Administrator's and Developer's Guide for E-mail Campaign Manager 1.0 (for CMS 6.2)

  o Administrator's and Developer's Guide for E-mail Campaign Manager 1.1 (for CMS 6.3)

  o Administrator's and Developers' Guide for E-Mail Campaign Manager 1.2 (for CMS 6.4)

  o Administrator's and Developer's Guide for E-Mail Campaign Manager 1.3.3 (for CMS 6.5)

  o Administrator's and Developer's Guide for E-Mail Campaign Manager 2.0 (for CMS 6.6)