



Sitecore CMS 6 ECM Performance Tuning Guide

A developers 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

©2011 Sitecore Corporation. All rights reserved.

Table of Contents

Chapter 1	Introduction.....	4
Chapter 2	Identify Potential Performance Gains.....	5
2.1	Fragmented Indexes on the Analytics (DMS) Database	6
2.2	Bandwidth Limitations	7
2.3	CPU Utilization	8
Chapter 3	Tuning Procedures	10
3.1	Check ECM Version	11
3.2	Perform OMS Performance Tuning	12
3.3	Tuning	13

Chapter 1

Introduction

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

Sitecore has a track record, through existing customers, of obtaining daily performance values of 3,500,000 individually rendered and personalized emails per server on 8 core servers. This document will aid as the coarse-to-fine tuning guide of the ECM implementation to help realize high-volume peak performance in your environment. Using this guide, Sitecore has as track record of existing customers of increasing the performance by 200 pct., by adapting and optimizing the configuration to match the local conditions, surrounding infrastructure and environment of the ECM. As a benchmark, Sitecore has obtained a daily sending rate of up to to 3,500,000 individually rendered and personalized E-mails on a dedicated in-production server.

The chapters provided in this guide are as follows:

- Chapter 1 – Identify Potential Performance Gains
- Chapter 2 - Tuning Procedures

Chapter 2

Identify Potential Performance Gains

Bottlenecks and Common Performance Issues explore potential concerns that could decrease realized throughput between the Sitecore implementation and the MTA.

This chapter contains the following sections:

- Fragmented Indexes on the Analytics Database
- Bandwidth Limitations
- CPU Utilization

2.1 Fragmented Indexes on the Analytics (DMS) Database

ECM reads and stores information about every recipient as an Automation State record in the Analytics database. Because of this, it's critical for performance that the Digital Marketing System (DMS) be tuned to run at its optimal performance.

2.1.1 Solution

Following, is a list of possible solutions to fragmented indexes on the analytics database:

- To tune the DMS implementation, refer to the Sitecore OMS Performance Tuning Guide at: <http://sdn.sitecore.net/Reference/Sitecore 6/OMS Performance Tuning Guide.aspx>

2.2 Bandwidth Limitations

Understanding the bandwidth, or bandwidth limitations, that you have available between the Sitecore implementation and the MTA can be the difference between a successful high output E-mail campaign and an E-mail campaign that crawls along.

Sitecore is capable of generating large quantities of high quality E-mails optionally with attachments and / or embedded images at a high rate. For example: If the size of a single rendering (E-mail) is 100kbytes and you are trying to achieve a rate of 40 E-mails per second, the bandwidth between the sending server and the receiving MTA needs to be able to handle 32 Mbps (4MBytes/sec). Care must be taken to make sure that the available bandwidth can handle the load, or that the rendering size or your expectations are decreased to match the actually available bandwidth.

2.2.1 Solutions

The following is a list of possible solutions to bandwidth limitation issues:

- In low bandwidth situations, create text based E-mails with links to images. This can be accomplished by navigating to the Manager Root item and clearing the Embed images check box.
- Reduce the number of SMTP connections. This will reduce the number of E-mails that can be sent per second, so plan accordingly for longer times required to complete a campaign. However, doing so will reduce the likelihood of sending errors due to not enough available bandwidth.

2.3 CPU Utilization

CPU utilization is a key performance metric. It can efficiently be used 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 may indicate a potential performance improvement, namely, under and over utilization of the CPU. The former occurs when the average CPU utilization is low (below 50 pct.), and the latter occurs when the average CPU utilization is high (above 90 pct.). Each of these cases is addressed in the following two sections.

The targeted goal will be a 80 pct. CPU utilization, which still allows for the server to handel spikes.

2.3.1 CPU Over Utilization

CPU over utilization typically occurs when the load that is being placed upon the CPU(s) is greater than the capacity available. Commonly, this will be seen on a Sitecore CMS server that is not only handling content creation, editing, workflow and publishing, but is also tasked with the creation and sending of E-mail campaigns. However, this is not limited to the scenario of a shared server, since an improperly tuned dedicated ECM server can also experience CPU over utilization.

Solutions

- Throttle the CPU down by decreasing the number of MaxGenerationThreads. The MaxGenerationThreads determines how many threads will be processed simultaneously by the CPU. Refer to Chapter 3 – Tuning Procedures.
- Check to see if E-mail rendering times are slow (greater than 100ms) by viewing 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, refer to: Section - 2.3 Rendering Performance of the [CMS Performance Tuning Guide](#)
- Decrease the number of SMTP Connections, to reduce the number of E-mails being sent concurrently
- If your server is loaded with many tasks besides sending emails consider adding one or more dedicated servers to off load the creation and sending of campaigns. Refer to the “Configuring Dedicated Servers” section of the appropriate E-Mail Campaign Manager Administrator’s and Developer’s Guide. Based on the ECM version, the E-Mail Campaign Manager Administrator’s and Developer’s Guide can be found at:
 - [Administrator's Guide for E-mail Campaign Manager 1.0 \(for CMS 6.2\)](#)
 - [Administrator's Guide for E-mail Campaign Manager 1.1 \(for CMS 6.3\)](#)
 - [Administrator's Guide for E-Mail Campaign Manager 1.2 \(for CMS 6.4\)](#)

2.3.2 CPU Under Utilization

CPU under utilization typically occurs when you are not taking advantage of the amount of available CPU resources to create and send E-mail campaigns. Commonly this is found in the case of a dedicated server, used to create and send E-mail campaigns, where the CPU utilization is low (below 50%). A properly tuned ECM solution will allow you to gain peak performance while creating and sending E-mail campaigns.

Solutions

- Throttle the CPU up by increasing the number of `MaxGenerationThreads`. The `MaxGenerationThreads` determines how many threads will be processed simultaneously by the CPU. Refer to Chapter 3 – Tuning Procedures.
- Increasing the number of sending threads (`NumberThreads` setting).
- Increase the number of SMTP connections (`SMTP.MaxConnections` setting)

Chapter 3

Tuning Procedures

This chapter describes how to tune the ECM to achieve peak performance during the sending of a campaign to the MTA.

This chapter includes:

- Check ECM Version
- Perform OMS Performance Tuning
- MTA Emulation Mode
- Tuning Procedures - Shared Environment
- Tuning Procedures - Dedicated Environment

3.1 Check ECM Version

Sitecore highly recommends that the latest version of the E-mail Campaign Manager module be used. This will insure that any stability and performance enhancements will be included in the ECM implementation.

3.1.1 Procedure to Check ECM Version

To check the version the installed ECM module, do the following:

1. Navigate to the bin directory of the website. This is typically /Web Root/bin.
2. Right mouse click on the Sitecore.E-mailCampaign.dll file and click Properties.
3. Click on the Details tab.
4. The property that holds the version information is "Product version"

3.1.2 Sitecore Recommendation

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

3.1.3 Report Findings

Record the results:

The installed ECM product version = _____

The installed ECM product version is the latest available

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

The installed ECM product version is not the latest available:

Error. The results show that the latest ECM product version has not been installed. Sitecore highly recommends that the latest ECM product version be installed to take advantage of stability and performance improvements.

3.2 Perform OMS Performance Tuning

ECM works closely with the Analytics database. Performance issues that are related to the Analytics database quickly migrate over to performance issues with the ECM, so it is imperative that your DMS implementation be kept in tune.

Prior to tuning the ECM module, the DMS implementation must be tuned and healthy. Otherwise there is the possibility for false results from the ECM tuning effort.

For information on how to tune the DMS implementation to run at its peak performance, refer to the [OMS Performance Tuning Guide](#)

3.2.1 Sitecore Recommends

Sitecore highly recommends that DMS be tuned in accordance to the OMS Performance Tuning Guide prior to tuning the ECM module.

3.2.2 Report Findings

Record the results:

DMS has been tuned in accordance to the OMS Performance Tuning Guide ___Yes ___No

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

OK. The results show DMS has been properly tuned.

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

Error. The results show that DMS has not been tuned according to the OMS Performance Tuning Guide. Sitecore highly recommends that DMS be tuned prior to tuning the ECM module.

3.3 Tuning

This section describes how to tune the ECM implementation to achieve the maximum possible throughput, given the type of environment that you are running in. ECM environments are described as shared or dedicated.

- **Shared Environment** – For a shared environment, where the ECM is creating and sending E-mail campaigns on the same server with other CMS activities (creation of content, editing of content, workflow, publishing, delivery, etc), the CPU utilization wants to remain at a percentage that will allow for multiple activities to take place.

ECM - Targeted CPU Utilization: ~30-40 pct. during dispatchment.

- **Dedicated Environment** - For a dedicated environment, used solely for the creation and sending of E-mail campaigns, the CPU utilization wants to be maximized (but not over utilized) to get the highest possible output.

ECM – Targeted CPU Utilization: ~80 pct. during dispatchment.

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

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

In the example tuning in this document, the following is assumed:

- **Round-trip** – The round trip time between the ECM implementation and the MTA of 1 second. Information on how to calculate the target rate based on pinging the SCA MTA will be provided. Actual achieved sending rates may be higher or lower based on several factors, including but not limited to: bandwidth limitations, round-trip time, size and complexity of E-mails, time to render E-mails and caching of rendered output, system size and available resources, network speed, analytics database maintenance.
- **MTA emulation mode** – Testing will utilize the MTA emulation mode. How to configure and run an E-mail campaign in MTA emulation mode will be explained below.

3.3.1 Prerequisites

- **Round-trip** - Know the round trip time it takes to send an E-mail to the MTA, and receive a response back. This is important in determining the number of sending threads that may be required to achieve the performance goals you are looking for.
- **Bandwidth limitations** - Know your bandwidth limitations. If you are trying to send 32 Mbps (4MBytes/sec) worth of E-mails, but only have bandwidth to handle 8 Mbps, you will not reach your goals. In cases where you have limited bandwidth available, consider creating a text based E-mail with links over a rich E-mail with imbedded images. (This can be accomplished by navigating to the Manager Root item and clearing the Embed images check box)
- **CPU utilization** - Know the availability of CPU utilization during sending periods. This will differ for a shared environment from a dedicated environment. For a shared environment, optimal

conditions would be to schedule the sending of a campaign during periods when no other CMS activities (such as publishing) is taking place.

- **E-mail Campaign Manager** - Know how to use the E-mail Campaign Manager. This document does not describe the procedures for creating campaigns and E-mails, or scheduling and running them. For information on how to use the E-mail Campaign Manager, please refer to the administration documentation for the appropriate version in use.

3.3.2 MTA Emulation Mode

The ECM module provides a means of testing campaign throughput by emulating a MTA. MTA emulation allows you to mimic the round trip time required to send an E-mail from the Sitecore CMS to a MTA.

This section will describe how to configure the ECM module to run in MTA emulation mode.

Prerequisite

- In order to run in MTA emulation mode, the ECM module must have connectivity to a MTA. This can be a local MTA, remote MTA, or Sitecore Application Center (SAC) account. If a connection is not available, a "Failed to connect to SMTP Server" error will occur.

MTA Emulation Mode Configuration

The MTA configuration settings can be found in the Sitecore.EmailCampaign.config file:

```
<!--Specifies whether the message sending process is emulated without actual message
transmission via MTA.-->
<setting name="MtaEmulation.Active" value="true" />

<!--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.Active	Set to true to enable MTA emulation mode
MtaEmulation.MinSendTime	This is the minimum amount of time, in milliseconds, to emulate the sending of a single E-mail. Use this in conjunction with MtaEmulation.MaxSendTime to mimic the behavior of Sitecore CMS to MTA round trip time.
MtaEmulation.MaxSendTime	This is the maximum amount of time, in milliseconds, to emulate the sending of a single E-mail.
MtaEmulation.FailProbability	This setting allows you to introduce the probability of connection failure into your emulation.

Determining MinSendTime and MaxSendTime when using SAC

Measuring round-trip time (RTT) can be done by starting a timer when transmitting a packet to the server and stopping the timer when receiving the answer. This can be measured locally on the server running ECM using a ping command. The RTT contains the time to transmit a packet, process it on the server running ECM and receiving the answer for the Sitecore Application Center (SAC) server. [Note: If you are using a MTA other than SAC, substitute the address to ping as required.]

Setting the MinSendTime and MaxSendTime for MTA emulation can be accomplished by pinging the SAC server as follows:

1. Launch 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

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, please 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, which is probably available at the your ISP's web-site (the bandwidth is likely to be slightly higher than observed).

3. Using the following formula to calculate the bandwidth during min. and max.:

$$\text{bandwidth min} = 2 * (\text{psize1} - \text{psize2}) / (\text{min1} - \text{min2}) \text{ Bytes/sec}$$

$$\text{bandwidth max} = 2 * (\text{psize1} - \text{psize2}) / (\text{max1} - \text{max2}) \text{ Bytes/sec}$$
4. Estimate the size of the messages you are sending using ECM. For example, ECM message size = 12,800 bytes. ECM message size = _____ bytes
5. Enter the MinSendTime and MaxSendTime values in the configuration file using the following calculations:

The minimum amount of time 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 to emulate sending a single message (in milliseconds):
`MtaEmulation MaxSendTime = ECM message size [bytes] / bandwidth max [bytes/sec] / 1000 [sec/ms]`

3.3.3 Configuration Settings

The configuration settings can be found in the Sitecore.EmailCampaign.config file. The settings that will be focused on for tuning are as follows:

```

<!--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 max number of concurrent SMTP connections allowed
NumberThreads	This is the total number of sending threads that will be used for sending messages.
MaxGenerationThreads	This is the CPU throttle. It specifies how many sending threads can generate messages at the same time.
RecipientRequestSize	This is the number of recipients that are queued from the analytics database in a single request.
Sleep	This is the amount of time, in milliseconds, to wait between sending messages.

3.3.4 Tuning a Shared Environment

Tuning a shared environment, or server, involves making sure that there is enough CPU resources available to not only run an E-mail campaign, but also allow for the running of CMS related tasks. If your CMS server is already at its limits of CPU utilization consider running a dedicated ECM server to handle the load required by creating and sending an E-mail campaign.

Initial Configuration Values for Tuning

For demonstration purposes, the initial configuration values will be based on a sending rate of 40 E-mails per second.

Setting	Initial Value
SMTP.MaxConnections	Same as NumberThreads
NumberThreads	100

Setting	Initial Value
MaxGenerationThreads	<p>Equal to the number of available CPU cores. (Note: By default, the MaxGenerationThreads setting is commented and has a default value of Environment.ProcessorCount * 2)</p> <p>A value of the number of available CPU cores will approximate 50% CPU utilization, allowing for other CMS activities to run in the same environment.</p> <p>Depending on CPU architecture you might need 2*CPU cores to reach 50% CPU utilization.</p> <p>If You are in doubt of you servers processor capacity, please refer to the vendors web-page or use the following lines of to create a console application and evaluated the capacity.</p> <pre>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(); } } }</pre>
RecipientRequestSize	500
Sleep	0

Testing

1. Enable and configure the MTA emulation mode to mimic the round trip characteristics of your Sitecore CMS to MTA connection. See *section 3.3.2 MTA Emulation Mode* for information on how to enable and configure MTA emulation mode.
2. Create or use an existing E-mail that is a good representative of the size and rendering complexity of E-mail that you would typically send, as well as 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. In emulation mode, no actual E-mail will be sent.
3. Prior to Dispatching the messages, launch the **Task Manager** on the machine that is creating and sending the E-mails and click on the **Performance** tab.
4. Dispatch the messages and watch the CPU utilization.

5. Review the Understanding the Results below to find out where modifications need to be made, and retest.

Understanding the Results

To understand the results you will need to look at two pieces of information. The first is to calculate the sending rate, and the second is to see how the CPU utilization performed during the running of the E-mail campaign.

Calculate the sending rate

The start and finish times of the E-mail campaign can be found in the Sitecore log file as follows:

1. Open up the Sitecore log file that is relevant to the testing that was just completed.
2. Search the log file for the term "EmailCampaign: Dispatch Message". You should find two entries related to the test (make sure that the time stamp correlates to when the test was run).
3. The first entry will show when the E-mail campaign started.
 - o **Example:** ManagedPoolThread #1 10:26:22 INFO EmailCampaign: Dispatch Message (NewsHotfix_semch3): Started
4. The second entry will show when the E-mail campaign finished. To calculate the send rate: $\text{number of recipients} / (\text{finish time in seconds} - \text{start time in seconds})$
 - o **Example:** ManagedPoolThread #1 10:29:08 INFO EmailCampaign: Dispatch Message (NewsHotfix_semch3): Finished

CPU is utilization is low and sending rate is not reached

In this case, the sending rate is too low. Do the following 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. Refer to: *3.3.2 MTA Emulation Mode*.
- Increase the number of MaxGenerationThreads to increase the number of concurrent threads that are processed.
- Increase the NumberThreads configuration value. (Note, 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, 90-95%, on a shared environment could mean that the amount of CPU resources used to send an E-mail campaign could limit the ability to run any other CMS activities. A targeted goal of 80% CPU utilization will allow for the server to handle spikes in the load.

If CPU utilization is above 90%, you will want to determine where the bottleneck is, and rectify the problem. Below is a list of possible causes and solutions that should be viewed in order.

- **MaxGenerationThreads** – Decrease the MaxGenerationThreads value to reduce the number of concurrent threads that are processed. For 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

CPU under utilization typically occurs when you are not taking advantage of the amount of available CPU resources to create and send E-mail campaigns. Possible solutions to increasing the CPU utilization include:

- **MaxGenerationThreads** – Increasing the `MaxGenerationThreads` value will increase the number of concurrent threads that are processed. The maximum value that `MaxGenerationThreads` should be increased to, for optimal performance, is $2 * \text{number of available CPU cores}$.
- **NumberThreads** – Increasing the `NumberThreads` value will increase the number of available sending threads. By doing so, your target sending rate will now be at a higher level. Bandwidth considerations need to be taken into account so that you do not flood the available bandwidth between the ECM implementation and the MTA.

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

3.3.5 Tuning a Dedicated Environment

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

Initial Configuration Values for Tuning

For demonstration purposes, the initial configuration values will be based on a sending rate of 40 E-mails per second.

Setting	Initial Value
<code>SMTP.MaxConnections</code>	Same as <code>NumberThreads</code>
<code>NumberThreads</code>	100

Setting	Initial Value
MaxGenerationThreads	<p>Equal to the 2 * number of available CPU cores</p> <p>A value of 2 * number of available CPU cores will approximate 100% CPU utilization, providing optimal performance.</p> <p>Depending on CPU architecture a value of 4 * cores might be needed to reach full CPU utilization.</p> <p>If You are in doubt of you servers processor capacity, please refer to the vendors web-page or use the following lines of to create a console application and evaluated the capacity.</p> <pre>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(); } } }</pre>
RecipientRequestSize	500
Sleep	0
Setting	Initial Value

Testing

1. Enable and configure the MTA emulation mode to mimic the round trip characteristics of your Sitecore CMS to MTA connection. See *section 3.3.2 MTA Emulation Mode* for information on how to enable and configure MTA emulation mode.
2. Create or use an existing E-mail that is a good representative of the size and rendering complexity of E-mail that you would typically send, as well as 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. In emulation mode, no actual E-mail will be sent.
3. Prior to Dispatching the messages, launch the **Task Manager** on the machine that is creating and sending the E-mails and click on the **Performance** tab.
4. Dispatch the messages and watch the CPU utilization.

5. Review the Understanding the Results below to find out where modifications need to be made, and retest.

Understanding the Results

To understand the results you will need to look at two pieces of information. The first is to calculate the sending rate, and the second is to see how the CPU utilization performed during the running of the E-mail campaign.

Calculate the sending rate

The start and finish times of the E-mail campaign can be found in the Sitecore log file as follows:

1. Open up the Sitecore log file that is relevant to the testing that was just completed.
2. Search the log file for the term "EmailCampaign: Dispatch Message". You should find two entries related to the test (make sure that the time stamp correlates to when the test was run).
3. The first entry will show when the E-mail campaign started.
 - o **Example:** ManagedPoolThread #1 10:26:22 INFO EmailCampaign: Dispatch Message (NewsHotfix_semch3): Started
4. The second entry will show when the E-mail campaign finished. To calculate the send rate: $\text{number of recipients} / (\text{finish time in seconds} - \text{start time in seconds})$
 - o **Example:** ManagedPoolThread #1 10:29:08 INFO EmailCampaign: Dispatch Message (NewsHotfix_semch3): Finished

CPU is utilization is low and sending rate is not reached

In this case, the sending rate is too low. Do the following 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. Refer to: *3.3.2 MTA Emulation Mode*.
- Increase the number of MaxGenerationThreads to increase the number of concurrent threads that are processed. Note, if the MaxGenerationThreads is set too high, then performance degradation may occur as a result of extensive thread context switching.
- Increase the NumberThreads configuration value. (Note, 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 too high

High CPU utilization on a dedicated server could prevent the server from being able to handle a spike in load. A targeted goal of 80% CPU utilization will allow for the server to handle spikes in the load.

If CPU utilization is above 90-95%, you will want to throttle down CPU usage. Below is a list of possible causes and solutions that should be viewed in order.

- **MaxGenerationThreads** – Decrease the MaxGenerationThreads value to reduce the number of concurrent threads that are processed. For a dedicated environment, normally reduce the

MaxGenerationThreads to somewhere between 2 * and 4 * number of CPU available CPU cores. Rerun the test after making the configuration changes.

CPU utilization is low

CPU under utilization typically occurs when you are not taking advantage of the amount of available CPU resources to create and send E-mail campaigns. A properly tuned ECM solution will allow you to gain peak performance while creating and sending E-mail campaigns.

Possible solutions to increasing the CPU utilization include:

- **MaxGenerationThreads** – Increasing the `MaxGenerationThreads` value will increase the number of concurrent threads that are processed. The maximum value that `MaxGenerationThreads` should be increased to, for optimal performance, is 2 * number of available CPU cores.
- **NumberThreads** – Increasing the `NumberThreads` value will increase the number of available sending threads. By doing so, your target sending rate will now be at a higher level. Bandwidth considerations need to be taken into account so that you do not flood the available bandwidth between the ECM implementation and the MTA.

3.3.6 Common Performance Enhancements

Normally, the bottleneck in a server sending E-mails is the available CPU. On both shared and dedicated environments you can increase the number of emails the CPU can generate per second. Using the same advanced Sitecore techniques and caching that you use to optimize a web-page:

- **Rendering** – Rendering an email is the same as rendering a web page, and can be CPU intensive if not created with performance in mind. To check to see if the rendering of an email is causing excessive CPU usage, rerun the test again using a very simple, small text based email. If the CPU utilization drops, then there is an issue with the rendering of the emails. Refer to the [CMS Performance Tuning Guide](#) and [Recommend Practices](#) for information on how to test and create renderings.

On both shared and dedicated environments you can add more dedicated or shared servers

- **Dedicated Server(s)** – If bottlenecks have been removed to lower CPU utilization, and target send rate is not being achieved, the addition of one or more dedicated servers should be considered. For information on how to setup and configure a dedicated server, refer to:
 - [Administrator's Guide for E-mail Campaign Manager 1.0 \(for CMS 6.2\)](#)
 - [Administrator's Guide for E-mail Campaign Manager 1.1 \(for CMS 6.3\)](#)
 - [Administrator's Guide for E-Mail Campaign Manager 1.2 \(for CMS 6.4\)](#)