Performance and Sizing Guide

HP Operations Manager 8.16 for Windows

Introduction

The biggest challenge when measuring the performance of distributed network and system management applications such as HP Operations Manager for Windows is not measuring the performance itself, but determining the parameters that affect the performance of the application.

The following types of parameters impact the performance:

- Static parameters
  - For example, disk space requirements, kernel configuration, and so on
- Dynamic parameters
  - For example, memory usage of HP Operations Manager for Windows processes and CPU utilization
- Network parameters
  - For example network usage and protocol overhead (for example, jumbo frames)
- Application related parameters
  - Number of operators working in parallel
  - Number of messages per second received by the management server
  - Type and complexity of messages to be processed by the management server
  - Number of active and history messages in the message browser
  - Number of messages in the database (active and history messages are stored in one table)
  - Number of custom message attributes in messages

Executive Summary

This section summarizes the scope of the executed tests and the recommendations.

Test Scope

The HP Operations Manager for Windows performance tests are similar in nature to the tests carried out for HP Operations Manager for UNIX and HP Operations Manager on Linux. This makes it easier to compare the performance of HP Operations Manager on different operating systems.

In addition, HP Operations Manager for Windows specific tests focus on, for example, comparing the HP Operations Manager for Windows MMC and web consoles.

Note:
The core components of the HP Operations agent can send messages at a higher rate than an HP Operations Manager management server on any platform can process. (This was verified with the logfile encapsulator process opcle, which uses the least resources.) This document therefore does not cover the agent performance.

The tests are described in detail in the section “Tests” on page 6. For your reference, the description of each test includes the metrics that were measured, the parameters that were varied, the test scenario, the test results, and any conclusions that can be drawn from the test results.

No service objects were present during the tests, as the service tree was outside the scope of the testing scenarios.

Hardware Recommendations

- Use a quad-core processor for the HP Operations Manager for Windows management server. (A dual-core processor with the same clock rate will only achieve half the message throughput into the database.)
- Select a CPU with a high clock rate to improve performance.
- There is no need for more than 4 GB RAM on a split[1] HP Operations Manager for Windows management server because of operating system limits.
- Use a remote database, installed on a 64 bit system with 8 GB RAM, a dual-core processor, and fast disks connected via SAN. (The tests were carried out on a combined management server/database system. Additional tests showed smoother behavior when the database has access to its own hardware.)
- Make sure that the network connection between the HP Operations Manager for Windows management server and the database server is sufficient: low latency, GB/s.

Hardware Recommendations for Virtual Machines

- Apply the combined requirements for the HP Operations Manager for Windows management server, a remote database server and the host system itself.
- To compensate for host CPU usage, make the projected CPU clock rate 110%.
- The HP Operations Manager for Windows management server performance will be reduced because of the lower network and disk performance of virtual machines.

Configuration Options that Influence Performance

To improve performance, consider the following configuration options:

- Use as few custom message attributes as possible in messages.
- Open as few MMC consoles as possible.
- For MMC consoles, keep as few messages as possible in the browser to improve startup performance.
- Do not use more than three message stream interfaces on each agent and server.
- Only forward selected messages to improve processing performance on the server.

Configuration Options that Do Not Heavily Influence Performance

- Number of managed nodes, if they do not send any messages
- Less than 200 web consoles without filters
- Less than 500,000 messages in the database (web console only)

The capability of an HP Operations Manager for Windows management server is based on the number and types of messages that must be processed. The number of nodes depends on the setup of the management domain. The more processing HP Operations agents perform on the managed node (for example, local automatic actions or message filtering), the less performance impact can be observed on the management server. HP has tested and certified a maximum of 3,000 managed nodes per management server. It is not recommended to add more than 3,000 managed nodes to one management server.

[1] One computer running the HP Operations Manager for Windows management server without a local database, and a separate computer that hosts a database server.

Test Environment

Management Server Test Setup

The tests are based on HP Operations Manager for Windows version 8.16 with a Microsoft SQL Server Express 2005 database.

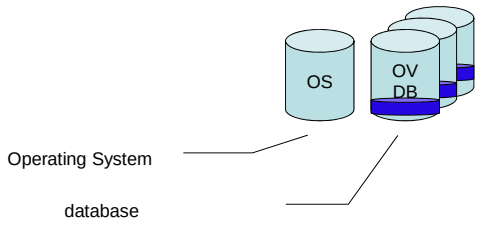
Table 1 Hardware Setup for the Management Server

Network	Dedicated 1 GB network for the consoles.
Machine Type	HP dl460c
Processor	4 x 2.83 GHz Quad-core: Xeon E5440
Main Memory	4 GB [2]
Disk Space	Disk array, via fiber channel (EVA 4000), sustained throughput 100MB/s
OS	Windows 2008
Swap space	Windows default
OS Patches	None

All disks were set up in the same way for all tests. The HP Operations Manager for Windows installation used a Microsoft SQL Server Express database (on the same system as the management server).

Figure 1 Disk Setup

The internal disk was used for the operating system and as a second disk for the database the EVA4000 disk array was used.



No special disk performance settings were used. Both disks were using standard NTFS. No additional operating system or product patch was applied. No part of the registry and no files were changed to enhance performance.

[2] The system has 8 GB but due to operating system restrictions, only 4 GB are available.

Console Test Setup

The following types of test systems were used to host HP Operations Manager for Windows MMC consoles.

Table 2 Hardware Setup for MMC Consoles

Machine Type	2 x HP dl460c
Processor	4 x 3.0 GHz Xeon Dual-core (Xeon E5450)
Main Memory	6 GB
OS	Windows 2003 Enterprise Edition R2

For the Web consoles, HP LoadRunner browser simulations were used.

Managed Nodes Test Setup

All nodes were loaded into the database in chunks of up to 2,000 nodes. The nodes were set up as Windows, 32 bit nodes with a name and an IP address. The nodes did not have any HP Operations agents installed.

Because of performance reasons, a hosts file was used instead of DNS.

Message Generation

HP LoadRunner was used to send single messages (no bulking).

To show maximum throughput, the local agent (opcle) on the server was used because the HP Operations agent can achieve better throughput by using bulk transfer (up to 1,000 messages consolidated into one chunk for communication with the server).

For the tests using bulk transfer, the local agent sent messages using its name as sending node. For other tests, HP LoadRunner generated specially crafted SOAP messages which were used to simulate HP Operations agents.

Management Server Settings

On the management server, message deduplication was disabled (unless otherwise stated). Apart from that, the default settings were used.

Tests

The following sections describe the executed tests in more detail.

Message Throughput without Connected Consoles

Table 3 Metrics and Parameters

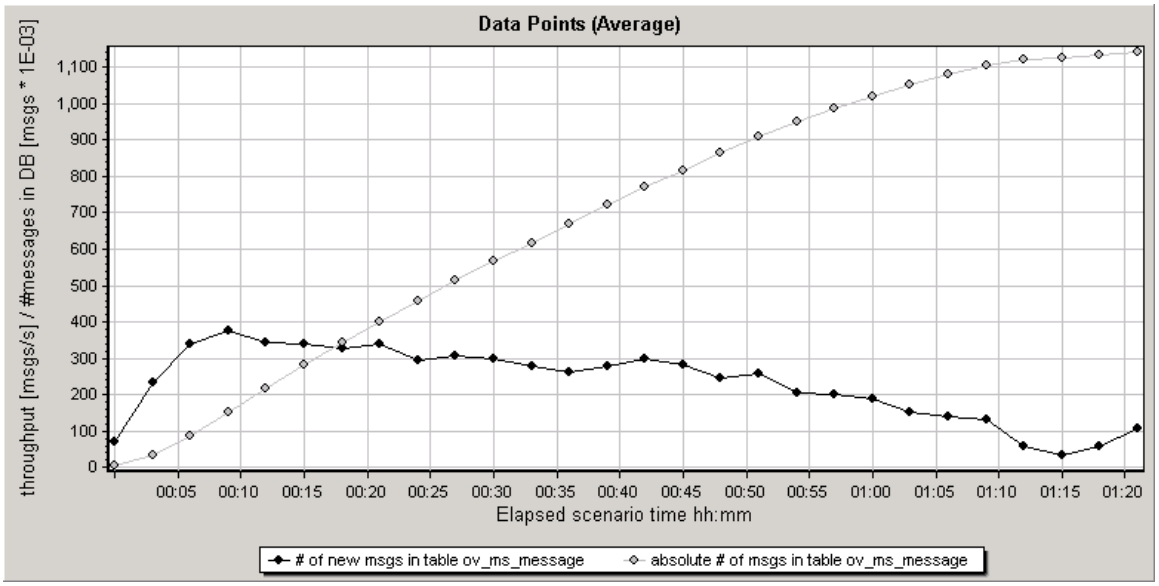
Metric	Number of messages per second (message throughput) to the database.
Parameter	Number of messages in the database.

Table 4 Test Scenario

Measured Value	Message rate into database at each point in time.
Message Generator	1,200,000 messages, severity normal with a rate of 1,000 messages per second for twenty minutes. Rate increasing slowly (by two messages per second) due to overload reaction of HP Operations Manager for Windows Database without any messages at the start of the test.

Test Results

Figure 2 Throughput example without connected consoles



The gray line shows the number of messages in the database (multiply by 1,000 to get real world numbers). The black line shows the throughput rate into the database (msgs/s). The figures show the slow increase of message delivery until 00:09 (soft start), and the slow decrease of throughput to the database until 1:10. At that time the server (mainly Microsoft SQL Server) started to swap.

Table 5 Test Results

Number of nodes	Rate with less than1,000 messages [msgs/s]	Rate with1,000,000 messages [msgs/s]
1,000	400	75

The following facts may be derived from these results:

- The message throughput decreases with the number of messages in the database.
- When Microsoft SQL Server can no longer extend its memory, a further decline in throughput can be observed. This does not apply when using a remote database.
- The number of nodes influences the throughput by reducing the amount of available memory.

Conclusions:

- The number of messages in the database has a big impact on the throughput. The throughput is stable up to 400,000 messages for less than 50.000 nodes (> 300 msgs/s in our tests). A remote database with more memory will keep the throughput stable with even more messages in the database.

Message Throughput Depending on the Number of Active MMC Consoles

Table 6 Metrics and Parameters

Metric	Number of messages per second (message throughput) into database.
Parameter	Number of MMC consoles.

Table 7 Test Scenario

Measured Value	Messages saved in database per second.
Message Generator	120,000 messages, severity normal. (Consoles can consume a lot of memory because of buffering mechanisms. More than 120,000 messages lead to an MMC memory issues (2GB), which also stalls message reception on the server.) Database without any messages at the start of the test.
MMC Console Options	Default configuration.

Table 8 Test Results

Number of MMC consoles	Throughput range	Typical rate
0	400	400
1	150-350	200
5	38-210	60
12	25-210	40
25	10-35	15

The following facts may be derived from these results:

- The more MMC consoles are connected, the slower the flow into the database.
- The more MMC consoles are connected, the more likely MMC will cause a freeze.
- Depending on the network connection and on the load on the client, there may be a short delay before the messages are seen in the MMC console.
- Connecting more than 25 MMC consoles is not advised.

Conclusions:

- Start only as many MMC consoles as you really need.
- If an event storm occurs, close as many MMC consoles as possible. (It is recommended to configure the HP Operations agent to prevent any message storms.)

Message Throughput Depending on the Number of Active Web Consoles

Table 9 Metrics and Parameters

Metric	Number of messages per second (message throughput) into database.
Parameter	Number of web consoles.

Table 10 Test Scenario

Measured Value	Messages saved in database per second.
Message Generator	1,200,000 messages, severity normal. Database without any messages at the start of the test.
Web Console Options	Default configuration.

Table 11 Test Results

Number of web consoles	Throughput maximum	Throughput minimum	Throughput minimum at number of messages
0	400	70	1.5M
5	380	70	1.2M
50	320	65	0.8M
300	300	50	0.2M

The following facts may be derived from these results:

- Web consoles do not have a strong impact on the throughput.
- Up to 200 consoles can be connected without problems.
- Memory issues (IIS) slow down throughput.
- Console response time is less than 10 seconds with 200 consoles even when all operators request new information every 30 seconds.

Conclusions:

- Use web consoles whenever possible instead of the MMC console.

Message Throughput with Deduplication of Messages

Table 12 Metrics and Parameters

Metric	Number of messages per second (message throughput) into the database.
Parameter	Message duplicate counter enabled or disabled. Number of different message keys (0, 1, 10,000).

Table 13 Test Scenario

Measured Value	Input flow rate into the database.
Message Generator	150,000 messages, severity normal, all messages were different (message text contained a unique number). No consoles. Database without any messages at the start of the test.
Miscellaneous	HP Operations Manager for Windows is configured to increment the counter only: no previous message was stored.

Table 14 Results Duplicate Message Suppression

Number of message keys	Throughput to database in messages per second (after reception of 150,000 messages)
No deduplication	400
1	430
10,000	401

The following facts may be derived from these results:

- The number of different message keys has a small impact on the performance (each new message key must be compared to the already existing ones: the more message keys there are already, the longer the time to process the new key).
- Taking into account that throughput decreases with the number of messages in the database, it makes sense to use deduplication to reduce the number of messages in the database.

Conclusions:

- If duplicate message suppression is enabled, message keys should be attached to the messages (even if each message has a unique message key).

Message Throughput with the Message Stream Interface Enabled

Table 15 Metrics and Parameters

Metric	Number of messages per second (message throughput) into the database via the message stream interface (MSI). Purpose of MSI: the original message text is appended with an MSI instance unique trailing text. MSI enabled on the server only.
Parameter	Number of MSI programs.

Table 16 Test Scenario

Measured Value	Database throughput rate using sqlcmd script.
Message Generator	1,200,000 messages, severity normal. Messages are targeted at different nodes. Database without any messages at the start of the test.
Console Options	No consoles were started.
Miscellaneous	All messages were **diverted** to one or more MSI programs using the server MSI. The MSI programs simply appended the MSI ID at the end of the message text. All MSI programs were registered with increasing IDs. The messages had to pass every MSI before being written to the database.

Table 17 Test Results

Number of MSIs	Throughput during message storm	Throughput when storm ended
0	400	400
1	25	80
2	10	80

MSI Programs: None
No MSI program was active; messages were processed directly without diverting, although forwarding to the MSI was turned on for the server.

MSI Programs: Server *n*xMSI
n MSI programs were active using the server message stream interface.

Message Throughput with Agent Bulk Transfer

In this test the messages were generated by the local agent on the HP Operations Manager for Windows management server. The messages were taken from a text file and converted line by line into messages through a simple policy. To provide maximum throughput nearly no processing was done by the policy (the line from the log file was converted directly into the message text). The internal mechanisms of the message agent (opcmsga) will bulk the messages received from the log file encapsulator (opcle) when a lot of messages arrive in a very short time. This leads to less overhead when transferring the messages to the server.

Table 18 Metrics and Parameters

Metric	Number of messages per second (message throughput) into the database.
Parameter	None.

Table 19 Test Scenario

Measured Value	Throughput into the database at any point in time.
Message Generator	1,200,000 messages, severity normal. Database without any messages at the start of the test.
Consoles	No console was connected to the server.

There were up to 700 messages per second entering the database.[3]

A queue was building up between the log file encapsulator (opcle) and the message agent (opcmsga). Another queue was building up before the OvEpMsgActSrv.

The following facts may be derived from these results:

- The log file encapsulator can create messages faster than the message agent can receive them.
- The message agent using HTTPS communication can deliver the messages faster than the OvEpMsgActSrv can process them.
- Bulking can speed up delivery into the database by up to 100%.

Conclusions:

- In case of a message storm, shut down the agents generating the storm.
  Any agent could fully load the server on its own.
  Remove the agent queues before reconnecting the agents.

[3] When HP LoadRunner simulated two messages in one bulk, up to 580 messages per second were observed (an increase of 50% compared to non-bulking).

Console Startup Time Depending on the Number of Nodes and Messages

Table 20 Metrics and Parameters

Metric	Startup time of MMC console. Startup time of web console.
Parameter	Number of managed nodes. Number of active messages.

Table 21 Test Scenario

Measured Value	Time until the console is started and fully functional.
Message Generator	5,000 and 50,000 messages, severity normal, were generated before the test started. Operators are responsible for all events. 500 or 2,000 managed nodes set up in HP Operations Manager for Windows. No history messages.
Console Options	Default configuration. 50,000 active messages in the browser.

Table 22 Results MMC Console Startup Time lists the results of this test: the time in seconds needed to start the consoles. Only one console was open at a time.

The startup time of the web console depends only on the number of web consoles that are already active. When approximately 300 web consoles are open at the same time, it will take about 30 seconds before the messages are displayed.

Table 22 Results MMC Console Startup Time

Number of messages	Number of nodes	Events per seconds	Startup time in seconds
5,000	500	0	12
5,000	2,000	0	12
50,000	2,000	0	408
50,000	2,000	5	435

The following facts may be derived from these results:

- The number of nodes has no significant impact on the startup time.
- The number of messages in the browser has a significant impact on the loading time.
- The number of active messages has an impact on the startup time of the MMC console, although a significant increase is seen only with more than 5,000 messages.

Conclusions:

- The number of nodes does not have an impact on the startup time.
- If a message storm occurs, use web consoles.

Other conclusions without showing numbers:

- The MMC console startup is faster than the web console startup if only few messages are kept in the browser.

- The web console startup time does not depend on the number of messages in the browser (number of messages displayed is much smaller anyway – only these will be read out of the cache and transferred into the browser). The startup time depends heavily on the local client and is therefore not shown.

Message Throughput Depending on the Number of Custom Message Attributes

Table 23 Metrics and Parameters

Metric	Message rate into the database.
Parameter	Number of custom message attributes attached to the messages.

Table 24 Test Scenario

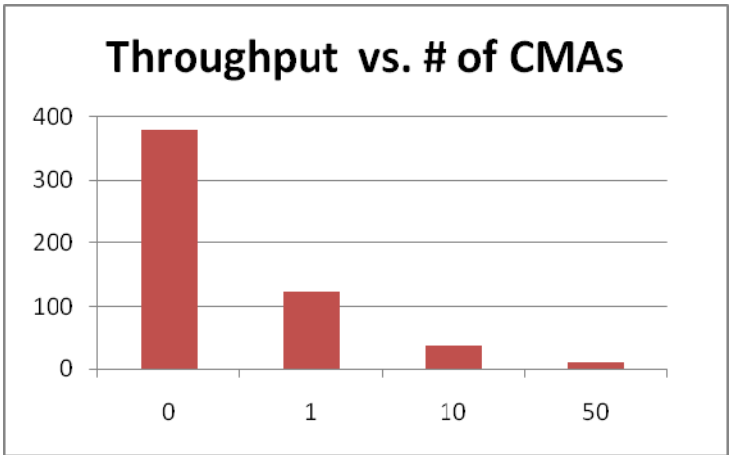
Measured Value	The input rate into the database when 150,000 active messages are already in the database.
Message Generator	Sending messages at maximum rate. Four tests were done with 0, 1, 10, and 50 custom message attributes attached to the test messages. No history messages for this test.
Connected Clients	No GUI was opened during the tests

Table 25 Test Results lists the throughput into the database with the corresponding number of custom message attributes per message when there are 150,000 active messages in the database.

Table 25 Test Results

Number of custom message attributes per message	Throughput into database [messages per second]
0	380
1	124
10	37.5
50	10.4

Figure 3 Event flow into the database depending on the number of custom message attributes per message



The following facts may be derived from these results:

- The number of customer message attributes per message has a strong impact on the throughput to the database.
- As a rule of thumb: if the number of CMAs is greater than 1 the maximum throughput is divided by the number of custom message attributes.

Conclusions:

- Use as few custom message attributes as possible.

History Download Depending on the Number of Downloaded Messages

Table 26 Metric

Metric	The time to download a varying number of history messages.
Parameter	None

Table 27 Scenario Varying the Number of Downloaded Messages

Measured Value	This test measures the time needed to download a varying number of history messages. The download was done using ovowmsgutil.
Message Generator	100,000, 500,000, 600,000, or 1,100,000 messages were in the history. No active messages existed. No messages were generated during the test.
Console Options	No consoles were started for this test.
Environment	Automatic downloading of history messages was disabled.

Table 28 Test Results

Number of history messages	Time in seconds	Time per message in milliseconds
100,000	270	2.7
500,000	1080	2.16
600,000	2220	3.7
1,100,000	6840	6.22

The following facts may be derived from these results:

- The download speed is about 300 messages per second.
- Below 800,000 history messages, the download time seems to be linear.
- Above 800,000 history messages, there is a shortage of memory and download is much slower (non-linear). (This does not occur with a remote database.)

Conclusions:

- Make sure to keep the number of history messages in the database below 800,000 to keep the download speedy.

Maximum Number of Agent Connections

The intention of this test is to show the maximum number of agents that can be connected without connection problems.

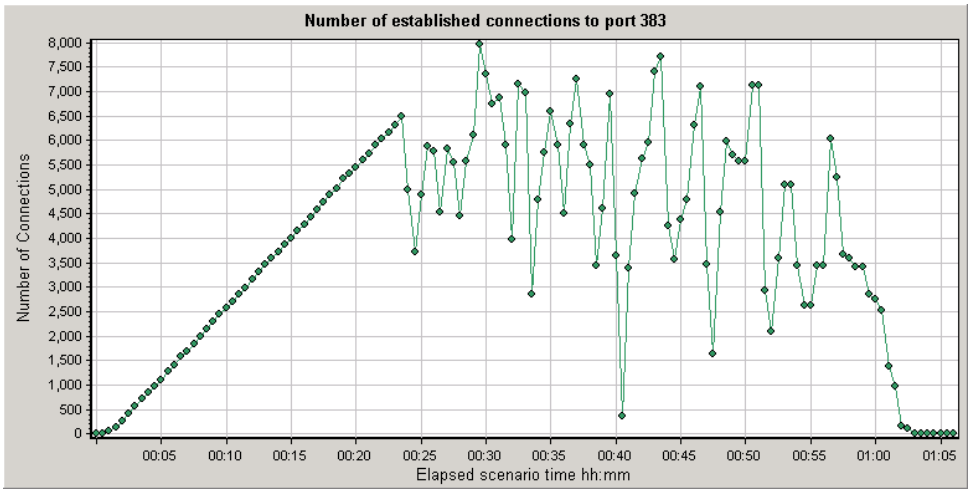
Table 29 Metrics and Parameters

Metric	Number of simulated agent connections to HP Operations Manager for Windows management server.
Parameter	None

Table 30 Scenario Maximum Number of Agent Connection

Measured Value	Increase the number of agent connections slightly over time. Each simulated agent will send a message after 119 seconds (TCP timeout is 120s).
Message Generator	Messages with severity normal. Message rate is increased to about 100 messages per second (12,000 simulated agents).

Figure 4 Number of established connections to port 383 on the server (8.16)



The following facts may be derived from these results:

- No problem for up to 6,500 connections. (A typical managed node requires two connections to the management server.)
- Erratic behavior above that limit (random disconnections).

Conclusions:

- Do not connect more than 3,000 nodes simultaneously.

Receiving Messages through Web Services

In this test the messages were delivered through the web service interface. The messages were delivered using HP LoadRunner. This test was also carried out on a second server equipped with only a dual-core processor (other machine parameters were the same).

Table 31 Metrics and Parameters

Metric	Number of messages per second (message throughput) into the database.
Parameter	None

Table 32 Scenario Receiving Messages through Web Services

Measured Value	Throughput into the database at any point in time.
Message Generator	Messages were delivered as fast as possible with 200 simultaneous connections from a load generator machine.
Consoles	No console was connected to the server.

Table 33 Results Receiving Messages through Web Services

	Dual-core mgmt server	Quad-core mgmt server (default)
Message throughput [messages per second]	20	60

The following facts may be derived from these results:

- The web service interface is about 84% slower than the HTTPS agent connection.
- Using four processors instead of two will increase the throughput by 200%.

Conclusions:

- Use the HTTPS agent connection to send messages.
- Use a computer with at least four processors (or four cores).

Message Forwarding

A decrease in message throughput was discovered when using message forwarding with all agents sending messages to management server OM1, and OM1 forwarding all messages to management server OM2.

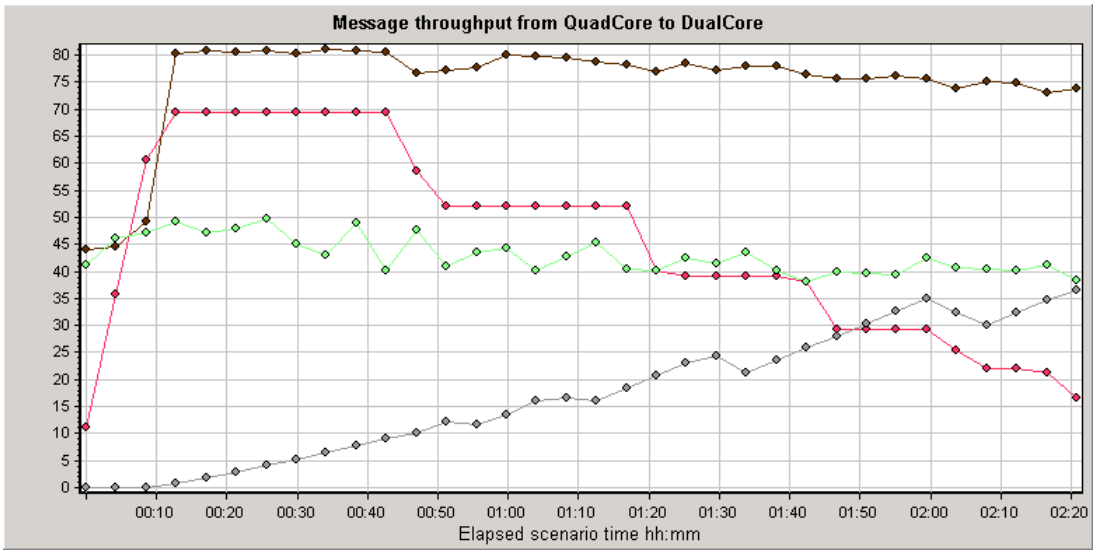
Table 34 Metrics and Parameters

Metric	Number of messages per second (message throughput) into the databases (OM1 and OM2).
Parameter	None

Table 35 Scenario Message Forwarding

Measured Value	Throughput into the databases at any point in time.
Message Generator	1,200,000 messages, severity normal. Delivery rate 80 messages per second. Database without any messages at the start of the test. OM1: quad-core (Xeon E5440), OM2: dual-core (Xeon E5150).
Console	No console was connected to the server.

Figure 5 Message throughput and queue size of management servers



Color	Scale	Measurement
	1E-07	Receiving MsgQueue on QuadCore
	1E-07	Receiving MsgQueue on DualCore
	1	Msg rate into DB on QuadCore
	1	Msg rate into DB on DualCore

The following facts may be derived from these results:

- Even on the quad-core server that receives the messages directly from the agents, the throughput is about 75 messages per second (as opposed to 400 messages per second on servers without message forwarding).

- DCE forwarding is no longer officially supported. Performance degradation compared to previous results (OVO 7.50) is caused by HTTPS security.
- Although the reception queue on the quad-core server decreases, message processing in OvEpMsgActSrv is still faster than message delivery into the dual-core server.
- Message reception on the dual-core server into the database is slower than delivery from the quad-core server (and additional processing there).

Conclusions:

- Use a computer with at least four processors (or four cores).
- The message rate in a flexible management environment will be about 20% of the value seen on the same computer with no message forwarding.

March 2011