Infor LN Sizing Guidelines Microsoft Reporting Solution

Contents

About this guide

This document provides performance guidelines, best practices and sizing information for using Infor LN 6.1 with the Microsoft Reporting Solution.

The Infor LN reporting solution covered in this document is:

- Microsoft SQL Server 2008 R2 Reporting Services (SSRS)

Intended audience

The intended audience for this document includes Infor professionals, Infor implementation partners and other specialists with a technical background.

Related documents

You can find the documents in the product documentation section of the Infor Xtreme Support portal, as described in "Contacting Infor" on page 7.

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at www.infor.com/inforxtreme.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

Chapter 1   Introduction

Overview

The Infor LN integration with Microsoft Reporting Services covers the printing of Infor LN reports using the Microsoft Report Server. This is called 'push' reporting, because the reporting data is pushed by Infor LN to the Report Server and not 'pulled' by the report itself.

The integration is available in Infor LN FP7 and higher feature packs.

The Microsoft Report Server software stack consists of:

- Microsoft Windows Server 2008 64bit
- Microsoft SQL Server 2008 R2 64bit
- Microsoft SQL Server 2008 R2 Reporting Services
- Infor LN Data Extension; this is the Infor LN plug-in for Microsoft SQL Server Reporting Services. There is a separate installer which installs this plug-in on the Microsoft Reporting Server.

The integration is supported both in Infor LN Web UI and Infor LN Worktop.

Supported types of reporting and report formats

Two types of reporting are supported:

- Report viewing; the report is rendered and displayed on the screen.
- Direct printing; the report is rendered and directly sent to an output device without viewing it on the screen. The output device can also be a file.

Both types of reporting support the formats CSV, PDF, MHTML, TIFF, Excel and Word.

Direct printing

With direct printing the report is sent to an output device without viewing it on the screen. Direct printing can be used for:

- Batch printing.
- Ad-hoc execution of (print) sessions.

- Report snapshots.

Direct reports can be sent directly to the printer or to a file. This last option is useful in case reports must be shared amongst many users. In this case the reports can be scheduled in a job and the results are stored in PDF format on a file share. Users are then directed to the generated PDF reports instead of executing the reports themselves.

With this approach you can create "snapshot reports" on a regular schedule. By using snapshots the reporting load on both the Infor LN server and the Reporting Server can be significantly decreased.

Example view of a HTML rendered report

The picture below shows an example of an HTML rendered report view in Infor LN Web UI. Once the report is displayed in the browser it is still possible to export the report to different formats, for example to PDF or Excel.
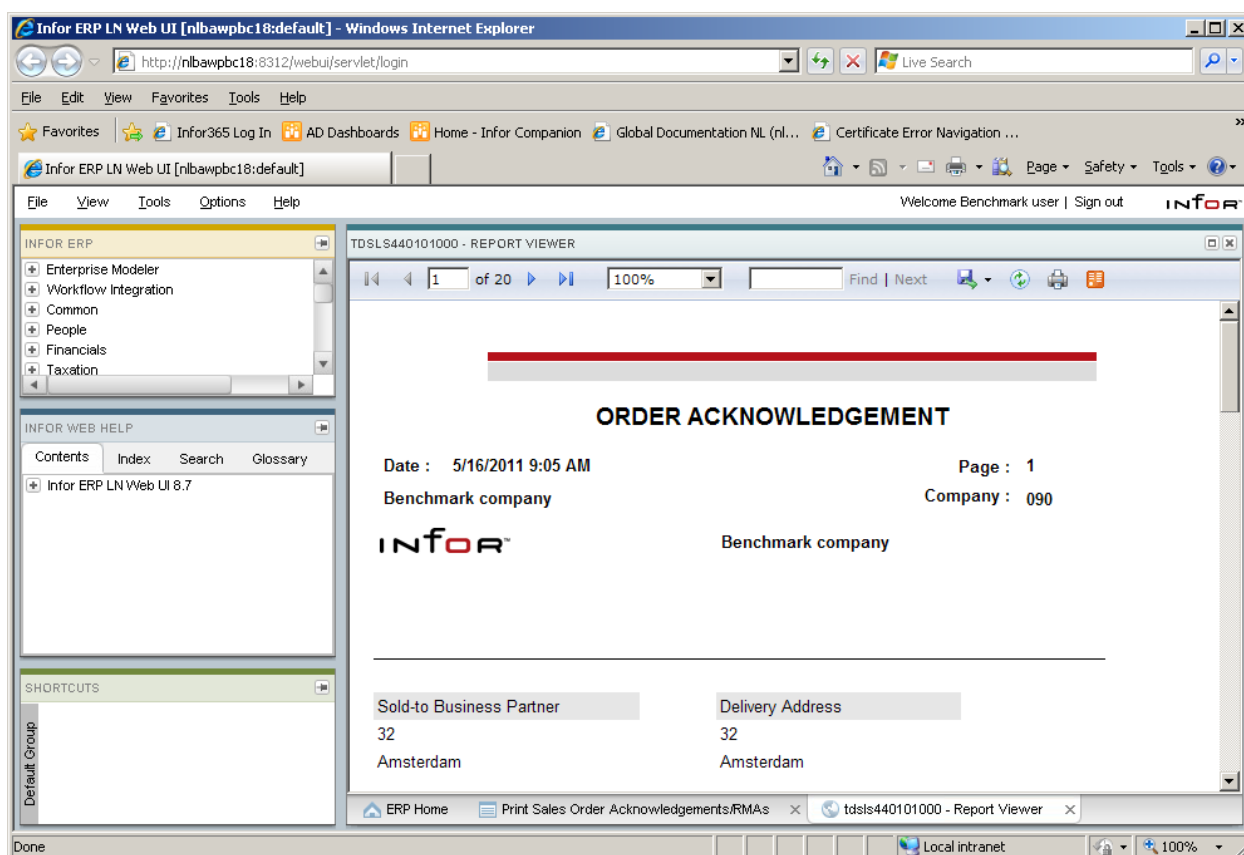
Figure 1 Example of an HTML rendered report in Infor LN Web UI

Chapter 2   Architecture

2

Introduction

This chapter describes the architecture of the Infor LN integration with Microsoft Reporting Services.

Report execution flow

When an Infor LN report is executed these steps are taken:

- An XML file is generated in the Infor LN temporary directory (%BSE_TMP%) containing the report data, this file is compressed.
- In case of report viewing:
    - o   A browser window is started.
    - o   In the browser window a Report Server request is executed to render the report. This request contains details on the report name and the location of the XML file.
    - o   The Report Server connects to the Infor LN server to retrieve the contents of the XML file.
    - o   The Report Server renders the report in the requested format.
    - o   The Report Server sends the rendered report back to the Infor LN server.
    - o   In case of report viewing in the HTML format: the first page of the report is shown in the browser.
    - o   In case of report viewing with different formats: the application, for example a PDF reader or Microsoft Excel, shows the entire report.
- In case of direct printing:
    - o   A Report Server web service is called to render the report. The web service request contains details on the report name and the location of the XML file.
    - o   The Report Server connects to the Infor LN server to retrieve the contents of the XML file.
    - o   The Report Server renders the report in the requested format.
    - o   The Report Server sends the rendered report back to the Infor LN server.
    - o   The report output is send to a printer device or a file.

Architecture of the Infor LN reporting solution

The picture below shows the runtime architecture of the MS reporting solution for Infor LN. Note that only push reporting from Infor LN is supported.
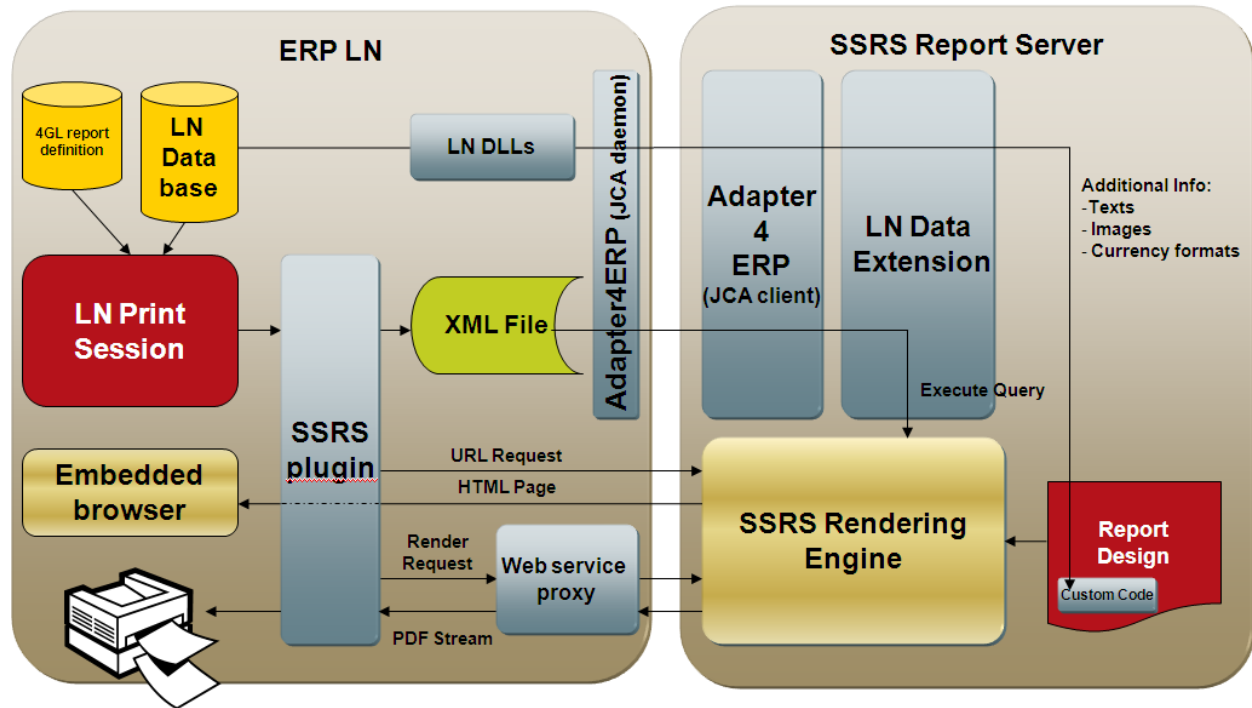
Figure 2 Architecture of the Infor LN reporting solution

The parts are described below:

LN Print Session / 4GL report definition / LN Database
The print session in LN queries the LN database and sends the data to the 4GL report. The 4GL report definition contains the so-called report input fields. These are the fields that finally are available for the Microsoft report.

SSRS Plug-in
This is a set of components in the LN backend that handle the printing of data with a Microsoft Report. It stores the available Report Servers, handles the flow of data when the user selects to print a Microsoft report and provides also some functions that can be called from the SSRS side to retrieve the (meta) data of the reports.

XML file
An XML file is created with the data to be printed on the report. In this file we have Report Properties, Labels and Data Rows. This file is used as the Data source on the SSRS side.

Embedded Browser / URL Request
When the user selected to view the report directly, an URL Request is build up and activated in the

Embedded browser of Web UI. At that moment SSRS renders the report and the HTML is send to the browser.

Render Request / Web service proxy / Printer
The user can also decide to print the report directly. In that case, a web service is called to render the report in PDF format. This PDF file is send to the printer.

Report Design / Custom Code
The report design is stored on the SSRS side in a SQL server database. This design specifies the data source to be used.

SSRS Rendering Engine
The rendering engine processes URL requests or web service requests. It executes the query that has been defined in the report design. For the LN push reports, this query retrieves the XML file from the backend. This is done by means of the LN Data Extension.

LN Data Extension
The LN Data Extension is an extension to the standard data source that SSRS can handle. This data extension retrieves the XML from the LN server (via the Adapter4ERP) and fills the internal dataset. SSRS uses this dataset to render the report.

Adapter4ERP
A common LN component that is used to connect a .NET environment to LN. It takes care of logging in on the server, starting a bshell and execution of BDE- and DLL-calls in LN.

LN DLLs
During the rendering of the report, it may be necessary to retrieve additional data from the server. This can be done from custom code in the report.

Chapter 3   Report server deployment

3

Introduction

This chapter describes several deployment scenarios of Infor LN using Microsoft Reporting Services.

The Microsoft Reporting Services contain these components:

- Report Server: Reporting services and Web services handling the report requests.
- Report Catalog: this is the database used by the Reporting Services.
- Reporting Data Source: In the Infor LN solution the Reporting Data Source is the XML data file generated from the Infor LN database.

Deployment scenarios

Minimum configuration

The minimum configuration to setup Reporting Services is to install a separate server (or Virtual Machine) hosting the Report Services and the Report Catalog (database).

LN Application Server   Report Server
LN Database   Report Catalog
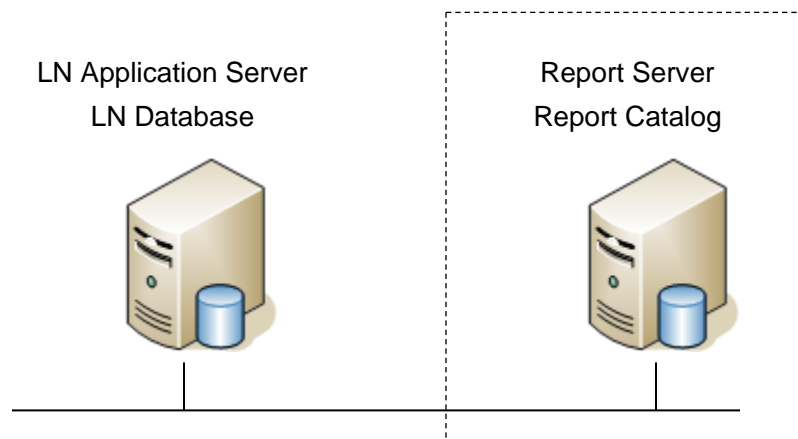
Figure 3 Single Report Server deployment

Centralized database server

In case the customer has a 3-tier Infor LN configuration it is possible to install the Report Catalog (database) on the Infor LN database server. Remember that Microsoft Reporting Services is part of Microsoft SQL Server, and that the Report Catalog needs to be stored in a Microsoft SQL Server database.

In this configuration there are more resources available for the Reporting Services because the Report Catalog activity is moved to the database server.
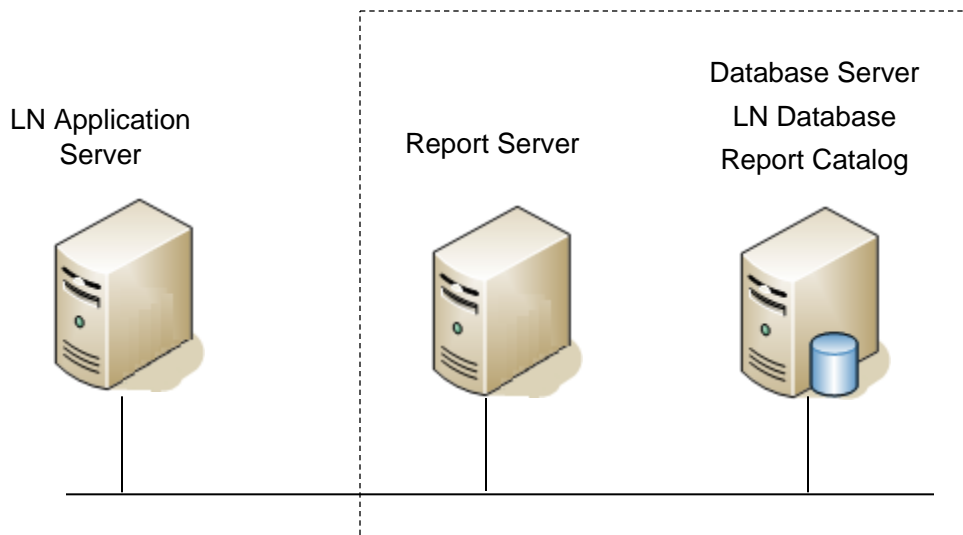
Figure 4 Single Report Server using a centralized database server

Report Server scale-out scenario

In case the reporting server capacity is insufficient it is possible to install multiple Report Servers, this is called a scale out scenario. Prerequisite for having multiple Report Servers is that the Report Catalog must be installed on a separate (database) server, because there is only once instance of the Report Catalog allowed.

Reporting Services does not provide functionality for load balancing a scale-out deployment or for defining a single point of access through a shared URL. You must implement a separate software or hardware network load balancer (NLB) solution to support a Reporting Services scale-out deployment. The network load balancer provides a single access point to the cluster of Report Servers, and balances the load over the Report Servers.

For performance reasons the use of a hardware network load balancer is recommended. For the Infor LN reporting integration it is required to affinitize the client requests to a single report server. This means subsequent requests of a user will always be handled by the same report server.
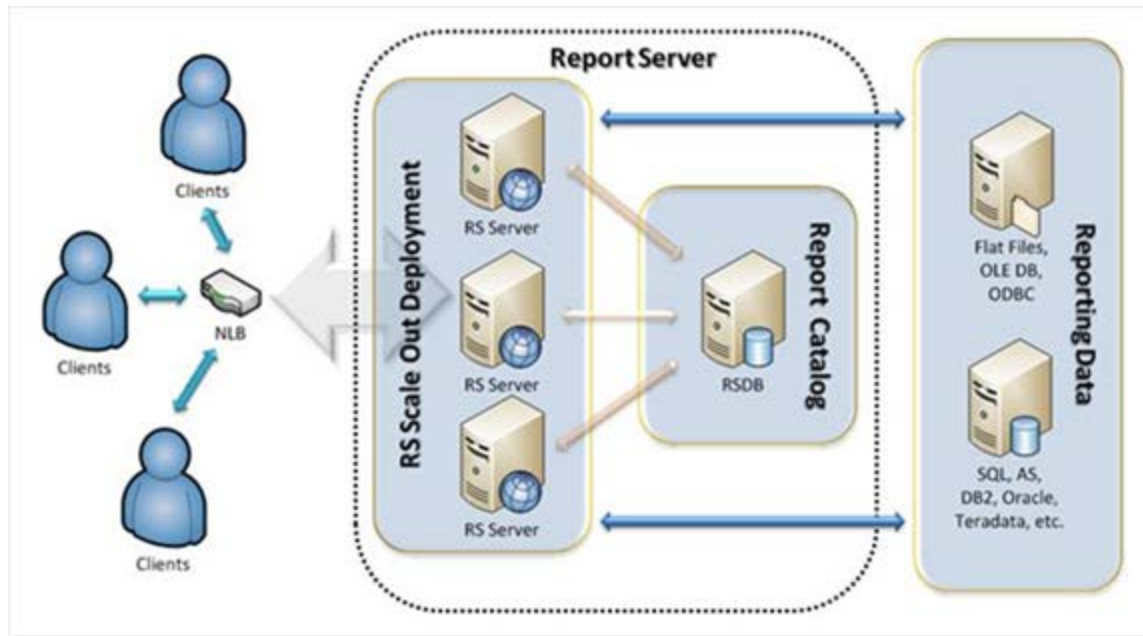
Figure 5 Generic Reporting Services scale-out scenario

The Report Server scale-out scenario for an Infor LN implementation is shown below.
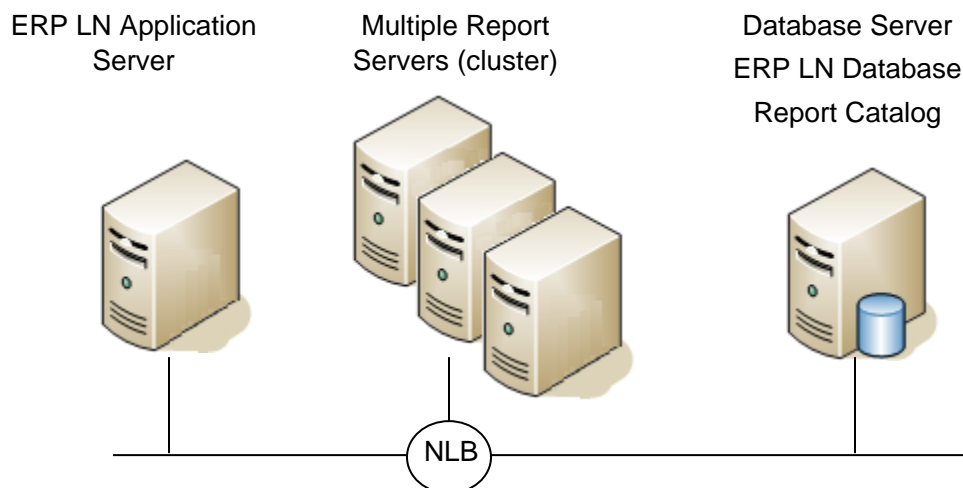
Figure 6 Report Services scale-out scenario for an Infor LN implementation

Special deployment scenarios

Multiple Infor LN application servers

Infor LN can be deployed using multiple application servers. In this case it is possible to use a single Report Server cluster for multiple Infor LN application servers, for example per region.
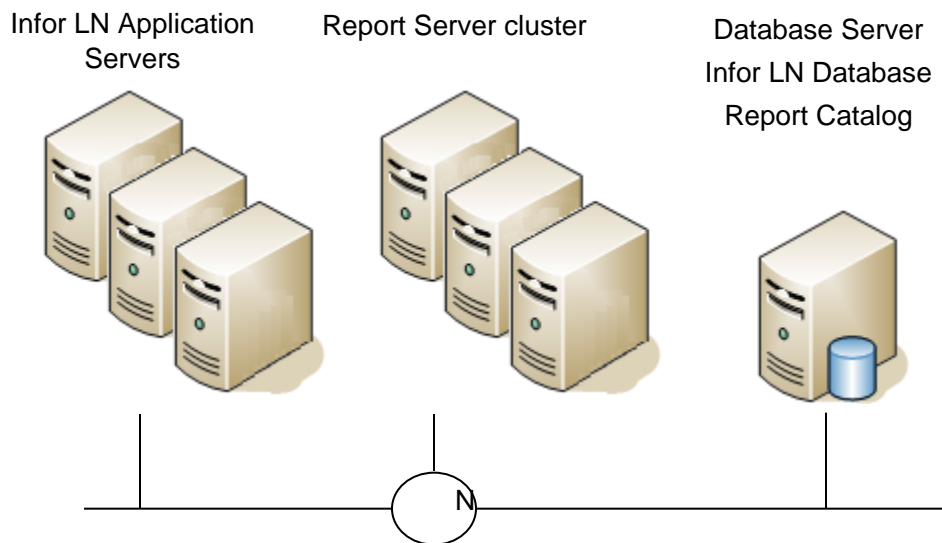
Figure 7 Multiple Infor LN application servers using a single Report Server cluster

There are some considerations in this setup:

- The report server needs fast local area network access to the Infor LN server. It is not recommended to use a wide area network between the Infor LN application server and the report server.
- The report server needs sufficient capacity to handle the reporting load of all the Infor LN application servers.

Chapter 4   Benchmark methodology

<div style="text-align:right">**4**</div>

Introduction

This chapter describes the business case and test scenario that are used to determine the sizing information in this document.

Benchmark setup

Reports tested

The Infor LN reports tested in the benchmark are:

Report	Report Type	Lines
Reprint Finalization Status	Simple	20 lines
Sales Order Acknowledgements	Complex	Small report: 100 lines
		Medium report: 200 lines
		Large report: 500 lines

The simple reports have a fixed data amount. The complex reports are further divided in small, medium and large data amounts.

A complex report is a report with significant data input and containing aggregations and calculations.

Benchmark scenario

The benchmark scenario is a mix of the two reports types and contains several report sizes:

Report	Report Format	Percentage in benchmark scenario
Reprint Finalization Status (tfgld1402m000)	HTML rendering	30%
Sales Order Acknowledgements/RMAs (tdsls4401m000)	HTML rendering	70% (of which 60% small reports, 20% medium reports and 20% large reports)

Business case

Each user executes the Infor LN reporting session and displays the first page of the report rendered in HTML. The benchmark scenario contains a mix of simple and complex reports with varying data sizes.

All reports are executed on live data; the caching and snapshot technology within Microsoft Reporting Services is not used, because for every report execution new data is pushed from the Infor LN server to the Report Server.

Reporting load

In the benchmark three load profiles are tested. The reporting load is expressed in number of reports per minute.

The table below shows the reporting load for each benchmark test:

	Reporting load		
Benchmark test	Simple reports / minute	Complex reports / minute	Total reports / minute
Test 1	7.5	17.5	25
Test 2	15	35	50
Test 3	22.5	52.5	75

Test environment

This chapter shows the hardware used in the benchmark tests. In the benchmark a single Report Server was used also containing the Report Catalog.

LN Application Server

LN Database

Report Server

Report Catalog

2 * Intel Xeon 5450
Quad Core 3.0GHz

16GB Internal
memory

2 * Intel Xeon 5540
Quad Core 2.53GHz

48GB Internal
memory

Users
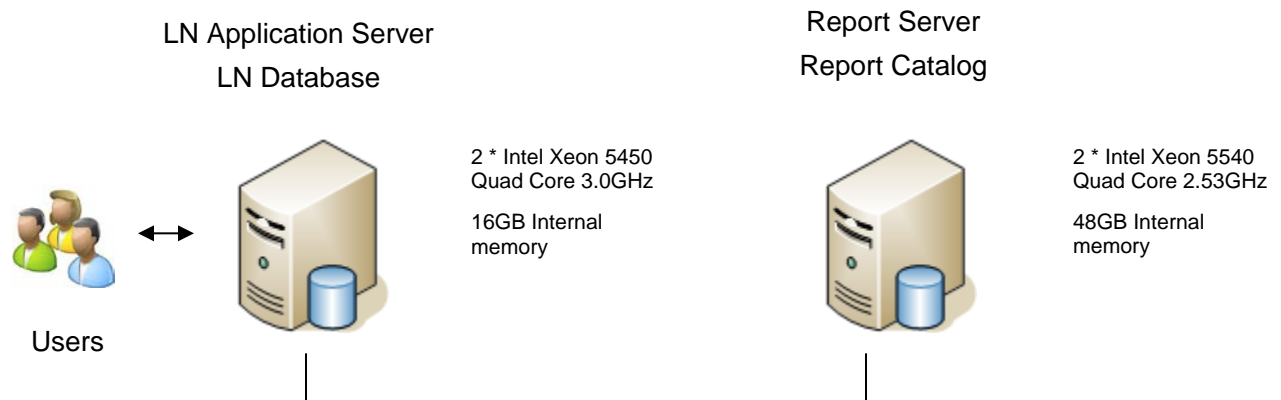
Figure 8 Benchmark environment

Chapter 5   Benchmark results

5

Introduction

This chapter shows the results of the benchmark tests executed at Infor.

Test scenarios and report server load

Refer to chapter 4 for more details on the benchmark methodology.

This table shows the tested scenarios and reporting load:

Benchmark test	Simple reports / minute	Complex reports / minute	Total reports / minute	
Test 1	7.5	17.5	=	25
Test 2	15	35	=	50
Test 3	22.5	52.5	=	75

The benchmark reporting load per scenario, and the actual number of Microsoft Report Server requests per second are shown below:
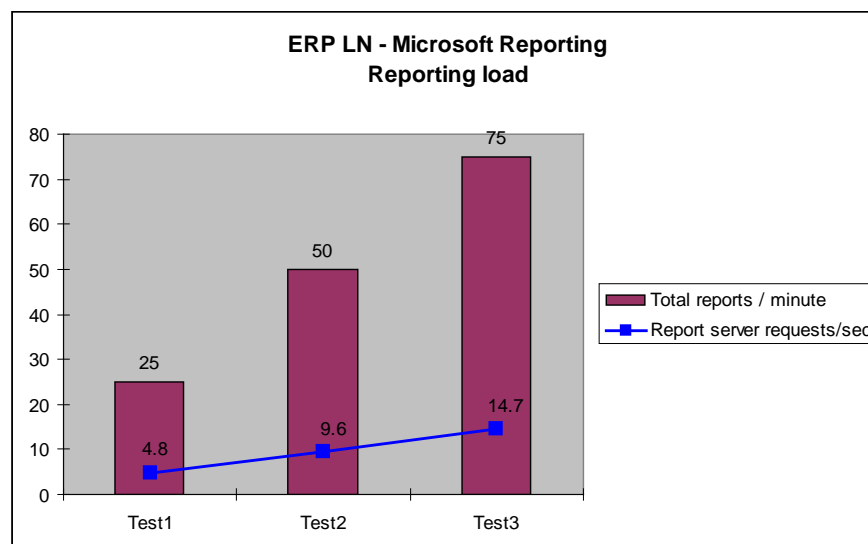
Response times

The average report response times for each scenario are shown below. There is a small increase in response time when the reporting load increases; this is due to the Infor LN server CPU getting saturated. The response times do <u>not</u> include client side processing time by the used browser.
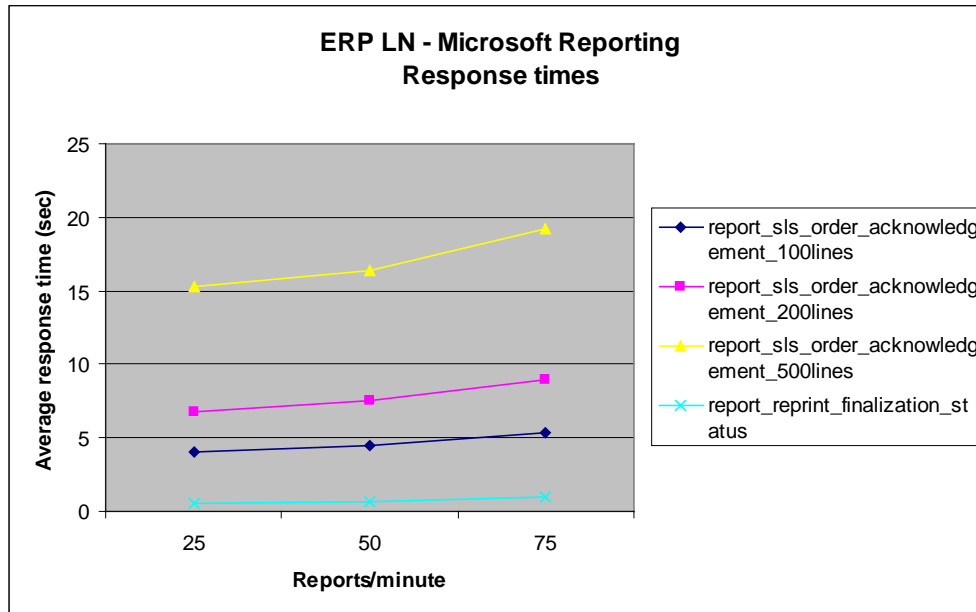
Resource usage

The <u>average</u> CPU usage on the Infor LN server and the Microsoft Reporting server are shown below. In the heaviest scenario there are CPU spikes up to 100% CPU usage.
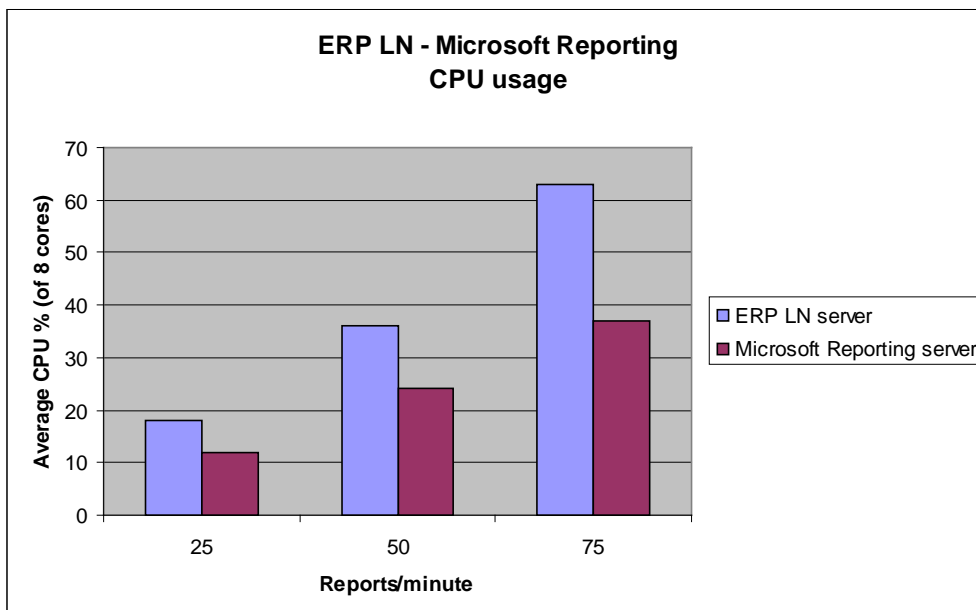
Disk IO

Disk IO occurs in these locations:

- In the Infor LN %BSE_TMP% directory for generating the temporary report files.
- In the Infor LN database for querying the report data.
- In the Report Catalog database on the Report Server.

The amount of IO in the benchmark was not very significant. The table below shows the benchmarked IO statistics on the Infor LN server and the Report Server (including the Report Catalog).

Average IO during the 75 reports/minute benchmark			
Infor LN IO per second	**Infor LN Kbytes/second**	**Report Server IO per second**	**Report Server Kbytes/second**
50	5000	100	1000

Chapter 6   Sizing recommendations

6

Introduction

This chapter contains sizing information for using Infor LN 6.1 with Microsoft Reporting Services 2008 R2. The sizing information is based on the benchmark tests described in this document.

The sizing discusses these parts of the reporting solution:

- Infor LN server
- Reporting Services server
- Report Catalog server (database)

It is often difficult to size for a reporting solution, because the reporting load is hard to predict and involves a number of factors:

- Number of concurrent reports being executed.
- Complexity of the reports.
- Amount of data processed and displayed on the report.

Using a separate server for reporting

Infor recommends installing a separate server or virtual machine for reporting. The reasons for this are:

- Reporting load is often unpredictable.
- Reporting load can have high peaks, e.g. at the end of the month.
- The impact of reporting on the Infor LN application performance must be minimized.

Most common performance bottlenecks

The most common performance bottlenecks of the Microsoft Reporting Solution are:

- CPU capacity of the Infor LN server. Each report uses CPU capacity on the Infor LN server to generate the reporting data.
- CPU capacity of the Report Server. Each report is rendered on the report server which consumes CPU capacity.

- Insufficient memory in the Report Server. Each report is temporary stored in memory. Under memory pressure Reporting Services will change its process model causing an increase in report response times.
- Disk IO capacity of the Infor LN database. This is because the Infor LN database is the data source for all the reports.

Concurrent users versus reporting load

The benchmark tests are based on a reporting load expressed in reports/minute. Depending on the reporting frequency this can be translated to a number of concurrent users.

In this table the benchmarked reporting load is translated to concurrent report users for a low, medium and high reporting frequency:

Benchmarked number of reports/minute	Reporting frequency		
	Low (1 report per user every 10 minutes)	Medium (1 report per user every 5 minutes)	High (1 report per user every 2 minutes)
25	250 users	125 users	50 users
50	500 users	250 users	100 users
75	750 users	375 users	150 users

Rule of thumb

Because of the high variance in the number of users accessing the report server at any point of time, the sizing must be based on the maximum number of concurrent users (i.e. the number of concurrent reports) that access the report server.

A general rule of thumb is to size the report server for 10% of the number of concurrent Infor LN users.

Example of a user based sizing

For example take an Infor LN implementation with 1000 concurrent users. Based on the "Rule of thumb" the amount of reporting users is estimated at 10% of 1000 = 100 reporting users.

The reporting load is estimated to be high (this is defined as 1 report per user every 2 minutes).

Based on the "Reporting Frequency" table above this translates to a peak reporting load of 50 reports per minute.

Minimum Report Server configuration

At minimum a separate server or virtual machine is advised with these specifications:

- 1 Dual core CPU
- 8 GB internal memory
- 3 disks in a RAID configuration
- 1 Gbps network

The estimated maximum reporting load for this configuration is 25 reports/minute.

Report Server sizing

Microsoft Reporting Services is memory consuming because each report's data set is placed in memory for performance reasons. The data in memory is for example reused when a user browses to the next report page. The more concurrent reports the more memory is being used. Therefore we recommend using a 64bit installation to handle a larger reporting load.

When there is insufficient memory available in the Report Server the file system will be used; in this case the reports will run slower and disk IO will increase.

The most common resource bottleneck in the Report Server is CPU, used for rendering the reports, and memory.

Based on the benchmark load this is the sizing recommendation for the Report Server including the Report Catalog database:

CPU	Maximum reporting load	Memory	Disk IO	Network
1 * Intel Quad Core	75 reports/minute (benchmarked)	16 GB	3 disks in a RAID configuration	1 Gbps
2 * Intel Quad Core	120 reports/minute (estimated)	32 GB	4 disks in a RAID configuration	1 Gbps

Report Catalog server sizing

Because all reports in the Infor LN integration are live reports the usage of the Report Catalog is limited. The Report Catalog will be used more in case of report caching and report snapshots.

In our benchmarks the amount of CPU used by the Report Catalog is less than 5% of the total CPU usage of the Report Server.

The Report Catalog consists of two databases:

- ReportServer; this database contains some metadata and the report definitions.
- ReportServerTempDB; this database contains snapshot data and history data.

The size of the ReportServerTempDB is determined by the number of concurrent reports. Each live report generates a report snapshot that is persisted in ReportServerTempDB for the duration of the user's session. This snapshot is e.g. used when you view the next page of a report.

For performance reasons it is recommended to install the Report Catalog database files on a RAID set of several disks.

For more information refer to the Microsoft SQL Server storage best practices:

http://technet.microsoft.com/en-us/library/cc966534.aspx

Infor LN server sizing

The load on the Infor LN server is composed by three parts:

- Running the database queries to get the report data. This step has impact on the Infor LN database.
- Generating the XML data file containing the report data. This step has the longest duration. During the generation of the XML file each Infor LN user uses a full CPU core.
- Delivering the data to the report server; the Infor LN server is the data source for the report. This step has a very limited impact; it only involves in transferring the data from the Infor LN server to the Report server.

In our benchmark tests the Infor LN server CPU (2 * Quad Core CPU) was close to saturation during the 75 reports/minute benchmark.

The amount of CPU used to generate the XML file was 70% of the total load on the Infor LN server. The querying of the report data in the Infor LN database was 30% of the total load on the Infor LN server.

All XML data files are generated in %BSE_TMP% on the Infor LN server. For large reporting loads it is important to make sure there is sufficient IO capacity on the %BSE_TMP% disk.

Based on the benchmark load this is the sizing recommendation for the Infor LN server. This sizing is additional to the standard Infor LN sizing:

Additional CPU	Estimated reporting load	Additional memory	Additional disk IO	Network
1 * Intel Quad Core	37 reports/minute	-	2 disks in a RAID configuration	1 Gbps
2 * Intel Quad Core	75 reports/minute	-	2 disks in a RAID configuration	1 Gbps

Remember that the IO increase is in %BSE_TMP%, for writing the temporary XML files, and in the Infor LN database, which is the data source for the reports.

Use of direct printing to PDF

While the figures in this document are based on viewing the first page of an HTML rendered report; some tests were executed to determine the differences with direct printing to PDF.

Direct printing involves more Report Server processing because the entire report needs to be rendered, and PDF rendering is more expensive compared to HTML. There is also an increase in CPU usage on the Infor LN server, because the rendered PDF report is directed to file by the Infor LN server.

The table below shows the increase in the benchmark scenario when using direct printing to PDF compared to viewing the first page of an HTML rendered report:

Increase when using direct printing to PDF	Factor
Average CPU increase on the Infor LN server	1.4 x (or 40% more)
Average CPU increase on the Reporting server	2.1x (or 110% more)
Average increase in report response times	2.2 x (or 120% more)

Chapter 7   Best practices

7

Introduction

This chapter describes several best practices for the installation and tuning of the Microsoft Reporting Solution. Best practices are described for both Infor LN and Microsoft Reporting Services.

Infor LN

Temporary report directory

The data for each report is generated in an XML file stored in %BSE_TMP% on the Infor LN application server. For large reporting loads it is important to make sure there is sufficient IO capacity and disk space on the %BSE_TMP% disk.

Tuning Microsoft Reporting Services

Maximum active requests

All reporting web service calls are executed by the same user account on the Report Server. By default the Microsoft Reporting Services is configured to support a maximum of 20 concurrent requests for a single user. It is recommended to increase this parameter to avoid reporting requests to be denied.

How to set the number of active requests

Modify the parameter MaxActiveReqForOneUser in the RSReportServer.config file. The default value is 20. It is recommended to increase this parameter to the number of Infor LN users. For example for a 200 user Infor LN installation:

```
<Add Key="MaxActiveReqForOneUser" Value="200"></Add>
```

For more information refer to this article: http://msdn.microsoft.com/en-us/library/ms157273.aspx

Reporting Services memory

By default Microsoft Reporting Services will use all available memory in the reporting server. It is possible to override this default behavior by configuring an upper limit on the total amount of memory allocated to Reporting Services.

When to customize memory settings

When there are running more applications on the same server and the Reporting Services takes too much memory, leaving insufficient memory for other applications.

How to set the memory limit

Add the parameters WorkingSetMaximum to the RSReportServer.config file. The value of this parameter is in kilobytes. For example, to set the maximum memory limit for Reporting Services to 4GB add:

```
<WorkingSetMaximum>4000000</WorkingSetMaximum>
```

For more details refer to this article: http://msdn.microsoft.com/en-us/library/ms159206.aspx

Auto growth settings for report server catalog databases

Microsoft Reporting Server installs two databases: the Report Server database and the ReportServerTempDB database. By default these databases are created with a small initial size and an auto growth setting of 1MB. It is recommended to increase the auto growth settings to avoid data file fragmentation and blocking during the auto grow task.

How to increase the auto growth setting

In the Microsoft SQL Server Management Studio select the properties of the database. Select files and update the auto growth setting for the data file to a minimum of 100MB for the Report Server database, and a minimum of 500MB for the ReportServerTempDB.

Update the auto growth setting for the log file of both databases to a minimum of 100MB.

How to increase the initial database size

In the Microsoft SQL Server Management Studio select the properties of the database. Select files and update the initial size setting for the data file to a minimum of 500MB for the Report Server database, and a minimum of 5000MB for the ReportServerTempDB.

Update the initial size setting for the log file of the Report Server database to a minimum of 100MB, and a minimum of 500MB for the ReportServerTempDB.

Troubleshooting

Error HTTP 503 Service Unavailable

This error occurs when the report server is unable to accept any new requests. A common cause is that the setting of the MaxUserRequestsForOneUser parameter is too low. Refer to the "Tuning Microsoft Reporting Services" section for more information on setting this parameter.

Report Server logging

The Reporting Server maintains logging information about report execution. In case of slow running reports investigate this logging information to determine the cause of the performance problem.

To retrieve information on report execution (ordered by starting time); run this query on the Report Server database:

```
select * from ExecutionLog2 order by TimeStart desc;
```

The columns TimeDataRetrieval, TimeProcessing and TimeRendering show where the time is spent for each report. The ByteCount and the RowCount columns show the size of the report. Example of two Infor LN reports (the timings are in ms):

	TimeDataRetrieval	TimeProcessing	TimeRendering	Source	Status	ByteCount	RowCount	AdditionalInfo
1	116	17	160	Live	rsSuccess	67047	41	<AdditionalInfo><ProcessingEngine>2</ProcessingE...
2	24	8	21	Live	rsSuccess	42814	42	<AdditionalInfo><ProcessingEngine>2</ProcessingE...

By clicking on the additional info link, you can see the actual amount of reporting server memory taken by this report. In addition, if the ScalabilityTime > 0 means that the Report Services is under memory pressure and will push long running reports to the file system. This is an indication that the Report Service has insufficient memory assigned. Example of the additional info, showing a report taking 98KB of memory and a server without memory pressure:

```
<AdditionalInfo>
  <ProcessingEngine>2</ProcessingEngine>
  <ScalabilityTime>
    <Pagination>0</Pagination>
    <Processing>0</Processing>
  </ScalabilityTime>
  <EstimatedMemoryUsageKB>
    <Pagination>13</Pagination>
    <Processing>98</Processing>
  </EstimatedMemoryUsageKB>
  <DataExtension>
    <INFORERPLN>2</INFORERPLN>
  </DataExtension>
</AdditionalInfo>
```

For more details refer to this article:
http://blogs.msdn.com/b/robertbruckner/archive/2009/01/05/executionlog2-view.aspx