

Performance Best Practices Guide for SAP NetWeaver Portal 7.0

Applicable Releases:

SAP NetWeaver 7.0

IT Practice:

User Productivity Enablement

IT Scenario:

Running an Enterprise Portal

Version 1.0

April, 2009

© Copyright 2009 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressly prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

Document History

Document Version **Description**

1.00	First official release of this document
------	---

Typographic Conventions

Type Style	Description
<i>Example Text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
Example text	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
Example text	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
< Example text >	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1.	Performance Best Practices	1
1.1	What is Performance?	1
1.2	Improving Performance	1
1.2.1	Analyzing Performance Problems	2
1.3	Server-Side Performance Factors	3
1.3.1	Improving CPU Consumption	3
1.3.2	Memory	4
1.3.3	Database Issues	6
1.3.4	Identifying Server-Side and Database-Related Problems	8
1.4	Network Performance Factors.....	9
1.5	Client-Side Performance Factors	10
1.5.1	Optimizing the Browser.....	10
1.6	Improving Performance on the Server Side	17
2.	Using the Navigation Cache to Improve Performance	19
2.1	Background Information	19
2.1.1	The Navigation Model.....	19
2.2	Configuring the Navigation Cache.....	20
2.2.1	Enabling/Disabling the Navigation Cache.....	20
2.2.2	Clearing the Connector Cache	21
2.2.3	Configuring the Connector Cache	22
2.2.4	Tuning the Connector Cache Configuration	22
2.2.5	Viewing Cache Content	23
2.2.6	Displaying Cache Statistics.....	24
2.2.7	Additional Cache Services	24
2.2.8	Determining the Cache Persistence Mode	26
2.2.9	Using Traces to Check the Cache	26
3.	Configuring Portal Runtime for Performance	27
3.1.1	Configuring the J2EE Portal Runtime Service	27
3.1.2	Configuring the Portal Properties.....	28
4.	Configuring the Portal Content Directory (PCD) for Performance	30
4.1	Overview of the PCD	30
4.1.1	PCD Structure	30
4.1.2	Unit Objects	30
4.1.3	Loading Time of a PCD Object	31
4.1.4	Factors Affecting Performance	32
4.2	General Guidelines and Tips	33
4.3	Configuring PCD Properties	34

4.4	Optimizing the PCD Cache.....	34
4.4.1	Background.....	35
4.4.2	PCD Cache Configuration Parameters.....	35
4.4.3	Recommended Configuration Settings for Cache Optimizer.....	36
4.5	Benchmarks.....	38
4.5.1	Fixed Test Parameters.....	38
4.5.2	Purpose.....	39
4.5.3	Results	39
5.	Page Content.....	44
6.	Hardware Sizing.....	45
6.1	Importance of Hardware Sizing to Performance	45
6.2	Sizing and Scalability.....	45
6.2.1	Understanding Scalability	45
6.3	Create Sizing Guidelines	47
6.3.1	CPU Consumption	47
6.3.2	Memory Consumption.....	47
6.3.3	Front-End Network Requirements	48
6.4	Apply Sizing Guidelines.....	48
6.5	Initial Sizing and Expert Sizing	49
6.6	Sizing Verification	50
6.7	References to Sizing Documentation	50

1. Performance Best Practices

The purpose of this guide is twofold: to help the system administrator to maximize the performance of SAP NetWeaver Portal and to provide the system administrator with starting points for initial analysis in the event that troubleshooting is required.

Note

For information about improving performance for Knowledge Management, see the guide [How To... Tune the Performance of Knowledge Management](#) on the SAP Community Network (www.sap.sdn.com).

1.1 What is Performance?

Performance may be defined by the following KPIs (key performance indicators) according to which performance is measured:

- Response time – Speed with which the system reacts to a request or other input
- Throughput – Number of successful transactions per time period, also known as the rate of work

Good performance is when these measurements satisfy your needs.

1.2 Improving Performance

After unsatisfactory performance is identified as a problem due to high response time or low throughput, it is necessary to determine the cause and the actions to be taken.

What You Can Do to Improve Performance

Actions to improve performance comprise any of the following:

- Tuning the system (requires understanding of the configurable parameters)
- Tuning the application
- Distributing load more evenly
- Optimizing code
- Adding hardware (sizing)

Identifying the Components in Which to Improve Performance

Before performance can be optimized you need to identify problematic areas. Performance involves a combination of many different aspects, but, in general, each performance problem can be categorized as belonging to one of following areas:

- Server
- Network
- Client

Note

You can use SAP Solution Manager for monitoring system landscapes, business processes, and for root cause analyses. More information: [Monitoring \(Overview\)](#) in the SAP NetWeaver Technical Operations Manual.

The following chapters explain the general concepts of each area above and provide ideas for analysis.

1.2.1 Analyzing Performance Problems

The high-level steps for analyzing performance problems are as follows:

1. First identify the software components involved in the business scenario and the interaction between them.
2. Identify the appropriate possibilities for monitoring the performance of the components.

Monitor the important metrics of every system involved in the scenario, for example:

- Application server
- Database
- LDAP
- Back-end systems

Monitor all processes:

- Dispatcher node
- Server nodes
- Database
- Enqueue server
- Message server

3. Reproduce the problem.
4. Conduct measurements and analyze the results.

Important metrics to gather for each process:

- CPU consumption
- Network load
- Memory consumption
- Paging
- Disk IO

Dealing with a Specific User Complaint

1. Identify the specific scenario.

Check if the performance issue applies to the entire organization or only to specific user groups or roles.

2. Check the hardware.

Is the user using a standard client machine?

Important

More information: See the SAP NetWeaver 7.0 Product Availability Matrix at:
<https://www.sdn.sap.com/irj/sdn/nw-70> → *Product Availability Matrix* → *SAP NetWeaver 7.0 Product Availability Matrix* → *Web Browser Platforms*

3. Gather information about the following:

- Number of HTTP round trips
- CPU usage – client and server
- Thread contention – server only
- Database usage (database locks, full scans, number of rollbacks)
- Memory usage – client and server

1.3 Server-Side Performance Factors

There are hundreds of parameters that can influence the performance on the server side. This section deals with topics that are more commonly known to affect performance, specifically: CPU, memory, garbage collection, and the database.

1.3.1 Improving CPU Consumption

The measurable KPIs for CPU consumption fall into one of the following categories:

- CPU consumption of a scenario or interaction step (during single user scenario) –For a single user performing a specific action alone and during load
- CPU utilization of a physical server (during load) – Multiple users are performing the same action.

1.3.1.1 Monitoring

There are a number of profilers on the market, all of which show CPU consumption to the granularity of a method. These are most suitable for monitoring single-user interactions.

Support tools with low overhead are more suitable for monitoring production systems. More information: SAP Note [797147](#) and the [End-to-End Root Cause Analysis](#) diagnostics page on SAP Support Portal.

1.3.1.2 Identifying CPU Problem During Load

During a load situation, if a response time of more than 3-4 seconds is reproducible, one of the following thread synchronization problems may be the cause:

- **Deadlock** – Indicates that threads are waiting for each other because of incorrect synchronization between them. This results in high response time but low CPU consumption.
- **Bottleneck** - Indicates that one thread is running in synchronized code for too long a time and all other threads are waiting for the same object, which is locked by the long-running thread. This results in high response time and high CPU consumption.

Follow this process:

1. Identify the problem by looking at the thread dump output, or by using any thread monitoring tool.
2. Trigger a few thread dumps at short intervals. More information: SAP Note [710154](#).
3. Find one of the following thread states, which may be causing a performance problem:
 - Waiting (for monitor / to lock monitor) – Thread is waiting for the release of a resource for a long time
 - Deadlocked – Where thread A is waiting for thread B and B is waiting for A

- Runnable – Thread is running on the same code for a long time, locking resources needed by other threads

You can use the Thread Dump Viewer tool, which show the thread dumps graphically, and can be downloaded from SAP Note [1020246](#).

More Information

The following SAP notes provide general and OS-specific information for the analysis of high CPU usage:

- General: [742395](#)
- Solaris: [743206](#)
- AIX: [743191](#)
- HP-UX: [743204](#)
- Windows: [743207](#)
- Linux: [743192](#)

1.3.2 Memory

The measurable KPIs for memory consumption fall into one of the following categories:

- Memory consumption of a scenario or interaction step (during single user scenario)
- Memory utilization of a physical server (during load)
- Percentage that garbage collection activity takes of total server time

1.3.2.1 Impact of Java Garbage Collection (GC) on Performance

Java garbage collection has a major influence on performance. Several collectors with different algorithms are available. These algorithms may differently implement their handling of young and old generation garbage collection, or the way they implement concurrent and parallel garbage collection. Each GC cycle runs through three sequential phases: mark, sweep, and compact. Regardless of different parallel and concurrent algorithms there is always a pause during which all threads are blocked. Intensive GC activity can cause poor response time, poor CPU utilization, and can endanger paging.

Acceptable GC Activity

To determine if the impact of GC activities is acceptable, prevent GC from consuming more than 5% of the total server activity time.

- Expected minor GC duration should be less than 0.2 seconds
- Expected full GC duration should be less than 10 seconds and should not occur at intervals of less than several minutes

Intensive GC activity, or even an out-of-memory situation, can be caused either by a memory leak or by high memory consumption.

1.3.2.2 Identifying the Memory Problem

Determining High Memory Usage

Use a telnet interface to explicitly trigger several consecutive full GCs.

1. Log on to the J2EE as an administrator (telnet <machine> <portxxx8>)



Note

Port must end with the numeral 8.

2. Type `1sc` to display cluster elements.
3. Jump to the server node.
4. Enter the `gc` command 3 or 4 times.
5. Measure the memory consumption before running the scenario you want to check.
6. Perform the scenario (for example, logon, navigation)
7. Repeat step 4.
8. Measure the memory consumption after running the scenario.

Check if the memory consumption reverts to near the original consumption or still stays at high levels. If the consumption is still high after the forced full GCs, check for memory leaks. Otherwise your scenario is consuming a lot of memory and this is the cause for extensive GC activity.



Note

To obtain lower level details of objects that are allocated in the scenario, use a profiler tool or a production monitoring tool, as described in SAP Note [797147](#).

Checking for Memory Leaks

Memory Leaks in the Java Heap

To investigate memory leaks, create a heap dump and use a memory analysis tool. The tool identifies large chunks of memory and shows what is keeping them alive.



Note

The memory analysis tool must support HPROF format of: Sun, SAP, HP JDKs/JVMs, from version 1.4.2_12 and higher, 5.0_7 and higher, 6.0 and higher.

See also the [Java Memory Analysis](#) page on SAP Community Network.

Memory Leaks in Permanent Space

When you receive an out of memory error, while your heap is not consumed to its maximum, the reason could be that there is not enough memory in the permanent space, which stores the bytecode of the Java classes in use. Configure permanent space size according to SAP Note [723909](#) via parameters `-XX:MaxPermSize=` and `-XX:PermSize=`

To check the consumption of permanent space you can trigger a GC via telnet, as described in *Determining High Memory Usage* at the beginning of this section, and look at the garbage collector output. An example of a line of GC output is as follows:

```
61.665: [Full GC 61.665: [Tenured: 0K->15021K(874496K), 0.4417612 secs]
94227K->15021K(1005056K), [Perm : 25503K->25503K(262144K)], 0.4419182 secs]
```

Note that the permanent space is shown as having 256MB, of which only 25MB are in use.



Note

The GC output format is relevant only for Sun, SAP, and HP JVMs.

Checking Memory Configuration

It is also possible that there is insufficient memory. Make sure that the available physical memory is utilized.

- Avoid paging for Java applications. **max. heap size + permanent space + stack** must fit into the available physical memory
- In rare cases, bugs in the GC itself can cause a performance problem. To prevent this use the latest VM version
- Wrong heap configuration can cause incorrect behavior of GC. For detailed Java VM Heap configuration, refer to SAP Note [723909](#)

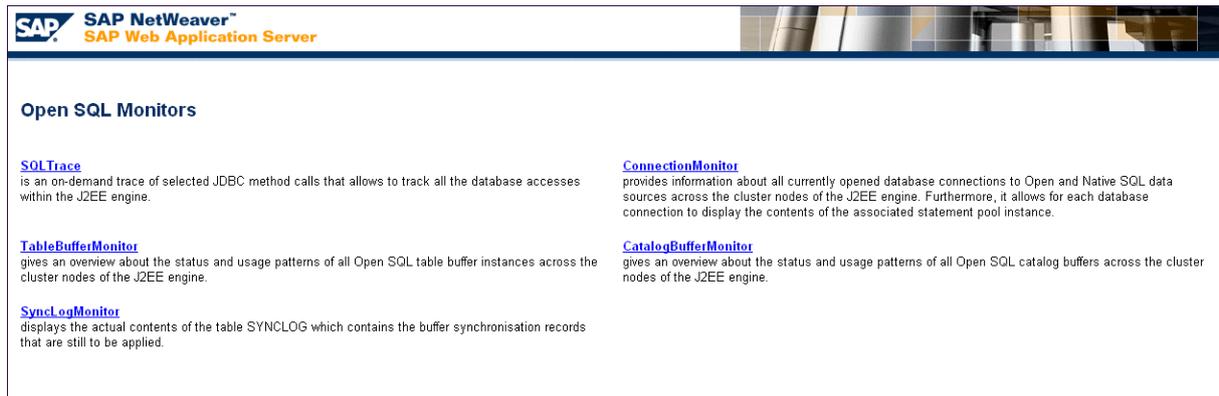
1.3.3 Database Issues

The database-related KPIs per user interaction step are as follows:

- Number of executed SQL statements (**select, insert, modify, delete**)
- Volume of transferred data

1.3.3.1 Using Open SQL Monitors to Measure SQL Queries Per Scenario Step

1. Go to `http://<host>.<domain>:<httpPort>/OpenSQLMonitors`.



Open SQL Monitors

[SQL Trace](#)
is an on-demand trace of selected JDBC method calls that allows to track all the database accesses within the J2EE engine.

[TableBufferMonitor](#)
gives an overview about the status and usage patterns of all Open SQL table buffer instances across the cluster nodes of the J2EE engine.

[Syncl ogMonitor](#)
displays the actual contents of the table SYNCL OG which contains the buffer synchronisation records that are still to be applied.

[ConnectionMonitor](#)
provides information about all currently opened database connections to Open and Native SQL data sources across the cluster nodes of the J2EE engine. Furthermore, it allows for each database connection to display the contents of the associated statement pool instance.

[CatalogBufferMonitor](#)
gives an overview about the status and usage patterns of all Open SQL catalog buffers across the cluster nodes of the J2EE engine.

2. Choose *SQLTrace* and, in the *Welcome to SQLTrace* screen, choose *Switch on and off SQLTrace* to open the *SQLTrace Status* screen.
3. Select the *Add backtrace* checkbox to include the call stack.

Recommendation

You can trace with or without performing a call stack; however, to identify the Java code responsible for calling a query, it is recommended to activate the SQL trace including the call stack.

4. Select the node or nodes to trace and choose *SQLTrace ON* to activate tracing per server node.

SQLTrace Status

SQLTrace ON

Refresh

Trace Evaluation

File Administration

Mon Sep 22 17:36:32 IDT 2008, running on node 903323150

Cluster Node	SQLTrace Status	Current Trace Id	Backtrace
903323150	Off		

Advanced Options:

Add backtrace (stack trace) to SQLTrace output

In case backtrace is enabled above:

Methodname contains (case sensitive)

Backtrace only in case method call takes longer than [microseconds]

5. Choose *Trace Evaluation* and set filters, if desired, to limit results.

Filters can also be set according to:

- Duration of queries (in microseconds)
- Session ID
- DSR Transaction ID
- User
- Application
- Min. Duration [microseconds]

6. Choose *Evaluate* to retrieve a list of the traces performed.

SQLTrace Evaluation: List for trace id 20080923132057957 from node 903323150.

Filter	List of Traces	Trace Status	Refresh		
<small>Running on node 903323150. Output limited to 5000 records in total.</small>					
Time	Duration [microsec]	J2EE user Application	Jdbc method Id	No. Result	Statement
13:21:01,744	380	Guest	PreparedStatement.executeQuery() STMT:6263824, RS:2721650		SELECT MIN("ITE_TIME") "ITE_TIME" FROM "CAF_GP_QUEITE" WHERE "QUE_LUID" = ? AND "ITE_STAT" = ?
13:21:01,746	3	Guest	ResultSet.next() RS:2721650	1 true	next()
13:21:01,746	6	Guest	ResultSet.close() RS:2721650		close()
13:21:01,746	2	Guest	PreparedStatement.clearBatch() STMT:6263824	1	clearBatch()
13:21:01,772	187	Guest	PreparedStatement.executeQuery() STMT:6263824, RS:25407975		SELECT MIN("ITE_TIME") "ITE_TIME" FROM "CAF_GP_QUEITE" WHERE "QUE_LUID" = ? AND "ITE_STAT" = ?
13:21:01,772	1	Guest	ResultSet.next() RS:25407975	1 true	next()
13:21:01,773	4	Guest	ResultSet.close() RS:25407975		close()
13:21:01,773	1	Guest	PreparedStatement.clearBatch() STMT:6263824	1	clearBatch()
13:21:01,773	313	Guest	PreparedStatement.executeQuery() STMT:5750104, RS:29178975		SELECT "ITE_LUID","ITE_TIME" FROM "CAF_GP_QUEITE" WHERE "QUE_LUID" = ? AND "ITE_STAT" = ?
13:21:01,774	4	Guest	ResultSet.next() RS:29178975	1 false	next()

7. Click a link in the *Statement* column to view details of the trace.

For example, clicking a *SELECT* clause shows the backtrace information: the classes and methods that called the query.

SQLTrace Record Detail (trace id 20080922172208236, node 903323150).	
Attribute	Value(s)
Location	com.sap.sql.jdbc.direct.DirectPreparedStatement.executeQuery()
Time	Mon Sep 22 17:22:16 IDT 2008
Duration [microseconds]	351
Statement	<pre> SELECT * FROM "CAF_GP_CODUE" WHERE "DUE_DATETIME" < ?₁ </pre>
Thread	SAPEngine_System_Thread[impl:5]_41
Bind Parameters	? ₁ = 1222093336767
ResultSetId	3815279
Table names	[CAF_GP_CODUE]
Backtrace information	<pre> at com.sap.sql.jdbc.common.CommonPreparedStatement.executeQuery(CommonPreparedStatement.java:185)at com.sap.engine.services.dbpool.wrappers.PreparedStatementWrapper.executeQuery(PreparedStatementWrapper.java:249)at com.sap.caf.eu.gp.model.co.execute.impl.CallableObjectExecutorPersistenceImpl.readOverdueTasks (CallableObjectExecutorPersistenceImpl.java:576)at com.sap.caf.eu.gp.model.co.execute.impl.CallableObjectExecutorTimeoutProcessor.process (CallableObjectExecutorTimeoutProcessor.java:158)at com.sap.caf.eu.gp.model.co.execute.impl.CallableObjectExecutorTimeoutProcessor.run (CallableObjectExecutorTimeoutProcessor.java:205)at com.sap.engine.frame.core.thread.Task.run(Task.java:64)at com.sap.engine.core.thread.impl5.SingleThread.execute(SingleThread.java:81)at com.sap.engine.core.thread.impl5.SingleThread.run (SingleThread.java:152) </pre>
Prepared Statement Id	9075814
VendorSQL statement Id	466728
DSR Transaction Id	d8b36f9088b111dda3f7001560a8db02

You can also download SQL traces in XML format for further evaluation (for filtering, sorting, and aggregations). More information: [SQL Trace](#) in the SAP Help Portal.

CAUTION

The SQL trace has a memory overhead. Be sure to deactivate it when it is no longer needed.

1.3.4 Identifying Server-Side and Database-Related Problems

Check the following:

- Number of times the database is accessed. Is it too high or inefficient?
- Number of simultaneous database connections
- Elimination of identical queries
- Appropriate table indexing defined for most commonly used `select` queries
- Database locks
- Correct functioning of load balancer
- Thread contention or thread deadlock
- Extensive I/O, that is, log writing due to exceptions or traces in debug mode
- Memory leaks or high memory usage
- Incorrect configuration of the Java heap (More information: SAP Note [723909](#))
- Incorrect configuration of OS

You can monitor database-related metrics for analysis. (More information: SAP Note [797147](#))

1.4 Network Performance Factors

The speed and capacity of a network are determined by latency and bandwidth. Bandwidth is the amount of data that can be transmitted in a fixed amount of time, usually expressed in bits per second (bps). Latency is the amount of time it takes a packet to travel from source to destination. Packet loss is another factor that can influence network performance.

The main KPIs to measure are:

- Number of round trips during a scenario – Each HTTP request and response cycle constitutes a round trip
- Volume of transferred data – HTTP headers and bodies make up the transferred data volume

In a typical Local Area Network (LAN) with an average 100 Mbps transfer rate, round trip time (RTT) is less than 10 ms. Neither the number of round trips, nor the transferred data volume has a significant impact on response time in a LAN.

In a typical Wide Area Network (WAN) with an ISDN or modem connection, RTT is about 250 ms (RTT for a satellite connection is about 500 ms). In a WAN, therefore, both data volume and number of round trips can influence the total response time significantly.

Improving Network Traffic

Use the following methods to improve network traffic:

- Caching
- Compression
- Reducing upstream and downstream data
- Reducing the number of requests

Recommendation

In many WAN landscapes, to overcome bandwidth and latency issues, it may be useful to consider the performance improving capabilities of Accelerated Application Delivery for SAP NetWeaver (AccAD).

For more information, see the AccAD page on the SAP Community Network, www.sdn.sap.com/irj/sdn/nw-accad.

Identifying Network-Related Problems – What to Check

- Volume of data transferred between client and server
- Volume of data transferred between server and DB
- Volume of data being transferred in one step of the scenario
- Number of round trips per step in a scenario, either between the front end and the back end or between two application servers

For optimal WAN performance, the number of round trips between the front end and the application layer per user interaction step should not exceed two on average.

Analyzing Network Connections

You can use the SAP NIPING program to help analyze network connections and metrics between machines running SAP applications. More information: SAP Note [500235](https://support.sap.com/en/notes/500235)

1.5 Client-Side Performance Factors

This section describes activities aimed at improving performance from the client side, such as browser and cache optimization.

Note

The information and examples in this section apply mainly to Internet Explorer 6 and 7; however, the general guidelines also apply to all browsers supported by SAP.

1.5.1 Optimizing the Browser

1.5.1.1 Browser Support

Verify browser compatibility. More information: <https://www.sdn.sap.com/irj/sdn/nw-70> → *Product Availability Matrix* → *SAP NetWeaver 7.0 Product Availability Matrix* → *Web Browser Platforms*.

1.5.1.2 Microsoft Internet Explorer Updates

Keep the browser up-to-date, not just with security updates but also with performance improvements and bug fixes that can enhance performance. For example, fixes that solve memory leak can prevent performance problems and speed up page processing.

Note

There is a patch published for Internet Explorer 6 that solves slow Web browser performance when browsing pages with JavaScript code. Tests show varied improvement from 0% and up to 50% depending on the exact scenario.

More information:

<http://www.update.microsoft.com/microsoftupdate/v6/default.aspx?ln=en-us>

<http://support.microsoft.com/kb/942840>

Avoiding Toolbars and Add-Ons

Toolbars and add-ons usually slow down page loading because they require resources or add more processing overhead. If you cannot avoid them completely, use them sparingly.

In Internet Explorer, go to *Tools* → *Manage Add-ons* and disable the items that you do not need.

Disabling the Pop-Up Blocker for the Portal

Some applications may use the pop-up blocker to close Web sessions. This depends on your pop-up blocking software. You can add the domain of your portal to the white list.

Reviewing Your Caching Configuration

The Microsoft Internet Explorer browser allows you to choose between several ways of checking for newer versions of stored pages. The default is *Automatically*, which is the recommended setting. Internet Explorer and the server share a common language and they negotiate about what to cache in the browser. The caching configuration that is set on the server decides and the browser follows the directives. Changing the algorithm to tell the browser to check for updates when there is no need causes more server requests. This involves more round trips and more server processing time. If it is urgent to verify that the content is the most recent, you can hit CTRL + F5 to reload the page from the server.

Recommendation

The recommended value of the disk space to use is 100-250 MB but you can estimate your cache requirements according to your business landscape.

Configuring the Recommended Browser Settings

Configure Internet Explorer options according to the following guidelines:

- Set to check automatically for new versions of stored pages
- Set disk space for the cache to a reasonable value (100 - 250)
Clear the cache after applying new settings.
- Allow the browser to use the HTTP 1.1 protocol

Recommendation

Consult your IT personnel before enabling HTTP 1.1 using a proxy connection. Using HTTP 1.1 via a proxy depends on the policy of your organization.

For advanced settings, we recommend allowing the browser to use the HTTP 1.1 protocol. HTTP 1.1 includes advanced cache headers and support of compression. Both are useful and can contribute to significant performance improvements. With this configuration your browser tells the server that it supports HTTP 1.1, but since it is the server that actually determines the protocol used, there are no compatibility issues.

Advanced Cache Settings

Some organizations do not allow keeping a Web cache on disk for security reasons. Because this policy can have a negative effect on performance, note that browser properties allow caching without saving sensitive data to disk. Check your settings. If caching is not enabled, consult your IT personnel.

- You can choose to enable or disable the saving of encrypted pages to disk (typically, content using the SSL protocol)
- You can choose to select an option that deletes temporary files, thus emptying the browser cache, when you close the browser

Many other browser parameters can affect performance, but the two parameters mentioned here are the most important. The parameters and their values may also depend on policies of the specific organization.

Investigating Client-Side Performance

You can monitor client-side activity by using your browser and HTTP tracing software. It is helpful to use a tool that can monitor the client-server round trips. It is best if the tool is a browser plug-in that can show what is loaded from cache and what is passed through the network.

Another option is a proxy-like tool that shows only what is going through the network. This also requires additional configuration.

Each tool is different but the basic principle is the same. You run the tool and perform your scenario—for example, log on to the portal and then stop the tool.

Afterwards you can analyze the data collected. The most useful information can be found in the request headers, for example:

- Number of round trips between client and server
- The amount of content (size of requests and responses)
- Cache access (which resources are loaded from the browser cache)
- Type of responses (HTTP codes)

Client-Side Tracing

Once it is verified that the browser is well tuned and patched with the latest fix, it is important to use a client-side tracing tool. This is a simple way to discover the many errors and problems that may arise, which would otherwise go unnoticed.

The primary goals are:

- Reducing the number of network round trips (utilize the browser cache)
- Reducing network traffic (make the request/response as thin as possible)

Choose a Simple Scenario

It is preferable to use a scenario involving a one-click navigation step as this makes it easier to isolate the problems.

Clear and Warm up the Browser Cache

It is better to record HTTP traces when the browser cache is full, as this is usually the common scenario. However, before you start measuring you should clear the cache once and perform your scenario to populate the cache. This removes old, no longer needed resources.

Tip

In Internet Explorer 6, if you browse to some page, clear the cache, and then click *OK*, the browser immediately reloads the page to cache; if you click *Cancel*, it does not load it.

Recording Your Trace

1. Go to the page you want to start from, for example the logon page, start your tracing utility, perform a navigation action in the browser, and stop recording when the page is fully loaded.
2. Save the trace so you can also reference it later for analysis.

Analyzing

1. First clear the obvious problems (usually errors).
2. Understand what is being called and repeat the trace to see if it is reproducible.

Response Codes

The common response codes are explained in the following table:

Response Code	Description
Cached	This resource is cached on your browser. The request is not sent to the server and you do not see this request in the HTTP logs. The browser fetches this resource locally from its internal cache.
	 Note Not all monitoring/tracing tools display a cache access parameter.
200 (OK)	This means the server behaves normally by sending back the resource you requested. If your resource is not cached, this is the preferred returned value.

Response Code	Description
302 (Found)	<p>This actually indicates a redirect, which happens when you request some URL and the server returns a new URL. The browser sends a new request to the newly received URL, which means a performance penalty of two round trips instead of one. Typically, this happens during logon where there is a redirect to some authentication service. It is preferable to avoid this situation, if possible.</p> <p>Example:</p> <p>If the value <code>Content-Length: 0</code> is returned in the header with the 302 code, it means that the response body is empty and only a URL is returned. The <code>Location</code> header indicates the URL.</p>
304 (Not Modified)	<p>When this value is returned, it is usually related to cache configuration issues.</p> <p>Even though the browser prefers to go to the cache before sending new requests to the server, the request header may contain a conditional request to verify if the cache is still valid. For example, the browser sends a request to the server for a resource, including a condition about the time the resource was last changed. If the server determines that the previous access to the resource is still valid it returns a header with the 304 code, without a body response. The browser then fetches the resource from its cache.</p>
401 (Unauthorized)	<p>User authentication either failed or is missing. This is normal for a first-time request before authentication. If you suspect a performance issue, you can check the URL that returned 401 and check the authorization issue.</p>
403 (Forbidden)	<p>Trying to view content for which the user has no authorization (even if the user has been authenticated). If you suspect a performance issue, you can check the URL and either extend permissions or remove this resource from the page.</p>
404 (Not Found)	<p>The resource is not located where you expect it and the server cannot find it. It may be a missing image that was not deployed or there may be a typographical error in the HTML. This is a problem that is usually easy to fix. Find the missing resource and put it in the correct location.</p>
500 (Internal Server Error)	<p>There is a problem on the server side. Generally, you can find the errors in the server log files.</p>
503 (Service Unavailable)	<p>This often means that the service you requested is still initializing and not fully started yet.</p>

 Note

You can find the complete list of HTTP response codes in RFC 2616 at <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>.

304 Response Code

This example considers typical data returned by a tracing tool. The two marked rows illustrate the 304 response code.

Started	Ti...	Time	Sent	Rec...	Me...	Result	Type	URL
+ 4.586		0.002	0	0	GET	(Cache)	image/jpeg	https://www.sdn.sap.com/fir
+ 4.592		0.000	0	0	GET	(Cache)	image/gif	https://www.sdn.sap.com/fir
+ 4.595		0.004	0	0	GET	(Cache)	image/gif	https://www.sdn.sap.com/fir
+ 4.598		0.002	0	0	GET	(Cache)	image/gif	https://www.sdn.sap.com/fir
+ 4.605		0.260	2161	393	GET	200	image/jpeg	https://www.sdn.sap.com/fir
+ 4.607		1.267	375	6704	GET	200	text/javas...	https://ssl.google-analytics.
+ 4.615		1.441	2365	538	GET	302	Redirect to...	https://www.sdn.sap.com/fir
+ 5.903		0.000	0	0	GET	(Cache)	image/gif	https://www.sdn.sap.com/fir
+ 6.129		2.245	1681	18290	GET	200	text/html; ...	https://weblogs.sdn.sap.cor
+ 8.397		0.275	1592	201	GET	304	text/css	https://weblogs.sdn.sap.cor
+ 8.732		0.717	1599	200	GET	304	image/gif	https://weblogs.sdn.sap.cor
+ 8.734		1.466	1600	200	GET	304	image/gif	https://weblogs.sdn.sap.cor
+ 8.735		0.006	0	0	GET	(Cache)	text/javas...	https://ssl.google-analytics.
+ 8.781		1.477	900	350	GET	200	image/gif	https://ssl.google-analytics.
		10.259	13724	32165	38 ...			

Request Resource Returned from Cache

The first marked row from the top is of type *image.gif*, a graphics file that the browser loads from cache, as shown by the *Cache* value in the *Result* column. The time estimation for retrieving it is 0.002 seconds.

Conditional Request Resource Returned from Cache

The second row, marked in the illustration, shows what happens when the cache is not sure if it has an updated version of a file. It sends an *if-modified-since* conditional request. The server receives the request and verifies the time stamp of the file and decides that the file did not change since the indicated time. The server now sends a 304 response code and the browser ends up retrieving the *image.gif* file from the cache. The total time for retrieving the file is 0.717 seconds, almost a full second for a small GIF file where both times the file was eventually loaded from the browser cache. (Since there may be parallel round trips, eliminating a 304 response code, in this example, might save between 0-0,717 seconds.)

Reasons for 304

Some of the reasons for getting a 304 response code when the resource is in browser cache can be avoided with proper configuration.

- Incorrect time settings on the server and/or client – The server or client machines are not configured with the correct time or time zone causing incorrect calculation of the expiration time.
- Browser cache algorithm – You can configure Internet Explorer to use one of several *Check for newer versions of stored pages*. If Internet Explorer is configured to check for a new version every time you visit a page it forces a 304 even if the page is still valid.
- Refreshing the page (F5) – Refreshing a page revalidates the versions of the resources even if they are still valid. This results in many 304 responses.

Tip

Using CTRL + F5 forces the browser to ignore the cache and reload the page, returning a response code of 200.

 Note

Sometimes resources that do not have any caching headers may be cached and the browser sends an `if-modified-since` request as an optimization. This depends on the browser algorithm.

Request and Response Headers

This section deals with the effect of headers on performance. It does not cover all headers or all the different combinations of headers, but discusses the most common ones.

The HTTP specification defines numerous header names that can be used. Most headers were defined in the 1.0 specification version but some were introduced in HTTP 1.1. Some headers are explicit, in which you specifically define behavior, and some are implicit, meaning that resulting behavior is dependent on some other header you defined.

The browser version and the presence of a proxy server also influence behavior. Proxy servers can manipulate and/or mask headers in order to enable providing their own functionality. It is possible to get one type of behavior if you go through proxy and another if you bypass it.

Usually, headers are separated by line breaks and are formatted as follows:

```
<header name>:<header value>
```

The HTTP specification defines which values are possible for each header name.

The following sections analyze and compare the more important headers, from a logon, of two examples of header sets.

Example 1: Server Request (Headers Received by the Server):

- `POST /irj/servlet/prt/portal/prteventname/Navigate/../../ HTTP/1.1`
`POST` or `GET` is the HTTP method used, followed by the requested URL and then the supported HTTP version (`HTTP/<supported version>`). The browser sends the highest HTTP version it supports. If 1.0 is indicated, check the browser settings.

 Note

It is also possible that you are using an application acceleration tool that does not use HTTP 1.1.

HTTP 1.1 offers new features, some of which are:

- Compression – Ability to compress data before it is sent to the browser, consequently lowering network traffic
 - Connection keep alive – Reuse of HTTP connections instead of opening a new physical connection for each request
 - Chunked response – The server can start sending data back to the browser even if it does not know the length of the full response
 - Caching - Improved cache handling headers
- `Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*`
 - `Referer: http://localhost:53000/irj/portal?..`
 Indicates who triggered this request – useful for tracing which file caused the request to be sent
 - `Accept-Language: en-us`
 - `Content-Type: application/x-www-form-urlencoded`
 - `UA-CPU: x86`
 - `Accept-Encoding: gzip, deflate`

The browser explicitly specifies that it supports `gzip` data compression, so that the server can compress the response body. It is up to the server to decide whether or not to compress. HTTP headers themselves cannot be compressed; the headers are needed to tell the browser that the body is compressed.

- `User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; .NET CLR 1.1.4322)`
- `Host: localhost:53000`
- `Content-Length: 298`
- `Cache-Control: no-cache`

In this case, the browser requests that the request-response not be cached. This is normal for POST since it involves submitting parameters that need to be processed on the server.

- `Connection: Keep-Alive`

The browser will reuse the connection after it receives the response, unless the response specifies otherwise (for example, `Connection: close`).

- `Cookie: PortalAlias=portal; saplb_*(ilperf125_F37_30)303540750; ...`

Upon each request, the browser sends a complete list of cookies relevant to the specific domain. Long cookies can be a major performance liability.

Server Response (Headers Received by the Browser):

- `HTTP/1.1 200 OK`

The server responds with the response code and the HTTP version it supports.

- `Server: SAP J2EE Engine/7.00`
- `Content-Type: text/html; charset=UTF-8`

Content type of the returned body

- `Content-Language: en-US`
- `Content-Encoding: gzip`

Content is compressed.

Most tracing tools can show the raw request and response. It can be helpful to see the uncompressed content rather than the compressed binary stream.

- `Date: Mon, 21 Jan 2008 15:29:24 GMT`
- `Transfer-Encoding: chunked`

Date and time at which the message originated

The server does not provide total response length.

Example 2: Response with Resources for Caching

- `HTTP/1.1 200 OK`
- `Server: SAP J2EE Engine/7.00`
- `Content-Type: application/x-javascript`
- `Last-Modified: Tue, 01 Jan 2008 09:40:19 GMT`
- `Cache-Control: max-age=86400`
- `Content-Length: 69729`
- `Date: Wed, 23 Jan 2008 15:44:59 GMT`

Comparing the Examples

The difference here is that the `Content-Type` header indicates that this is a JavaScript file and that the cache control returns with the value `max-age=86400`.

This means that the server tells the browser that it should cache the resource for 86,400 seconds (24 hours). For the next 24 hours the browser can load this file from cache whenever it is requested without the need to request it again from the server.

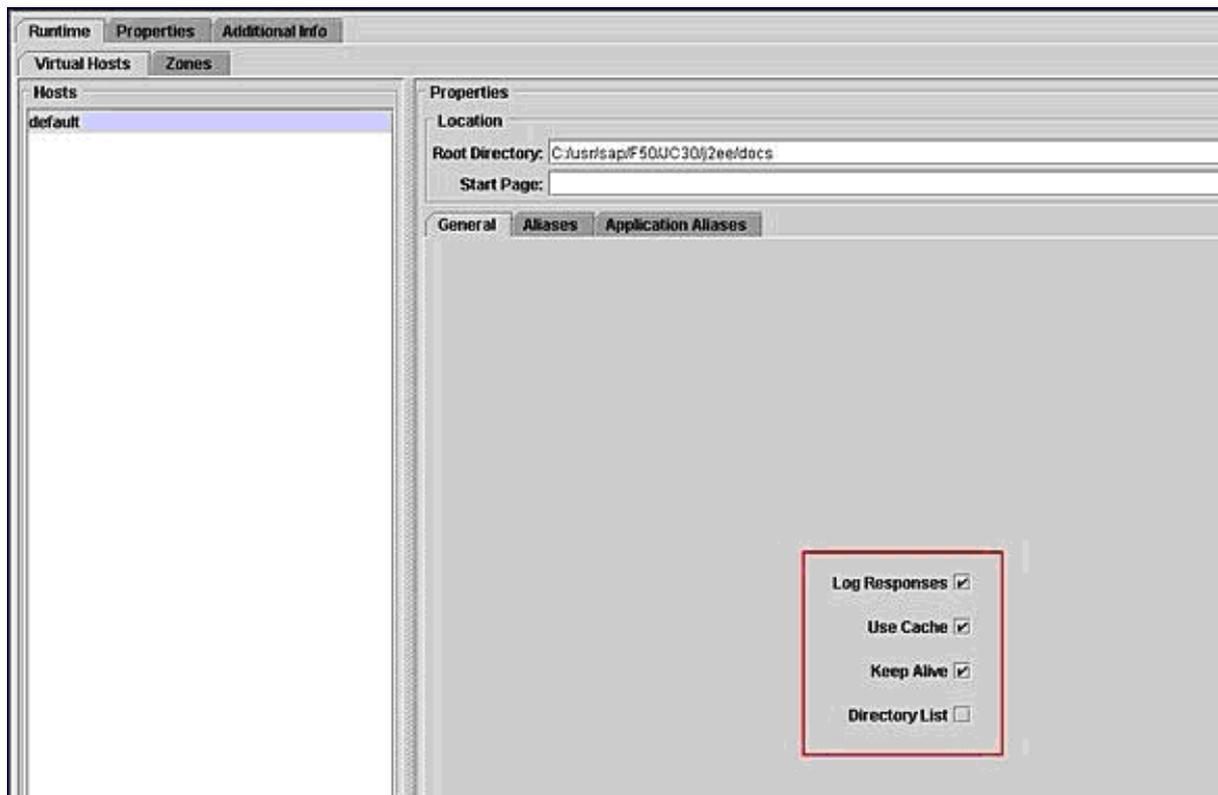
After 24 hours, the server sends an additional header, `Last-Modified`, which is the date the requested resource was last modified on the server. The browser saves this date, together with the cache definitions of the file, and when the cache period times out it sends a conditional request to the server asking if the resource was modified or not. If the resource was modified, the server sends the updated resource file to replace the cached version. If the resource was not modified, the browser sends a short response telling it to continue using the cached file and to continue to cache it for the next 24 hours.

1.6 Improving Performance on the Server Side

The HTTP server allows the configuration of many parameters. It is important to check the current configuration to verify that it matches your requirements.

1. From the Visual Admin tool of the application server, go to *Server_abc* → *services* → *HTTP Provider service*.

In the *Runtime* tab, you see the following four options, with checkboxes, which must be properly configured.



Option	Description
<i>Log Responses</i>	Enables logging of all requests. Performance is improved if you deselect this option. Logging is not necessary but be aware that the HTTP logs can generate useful statistical information.
<i>Use Cache</i>	Enables caching (selected by default)
<i>Keep Alive</i>	Allows connection reuse (selected by default)
<i>Directory List</i>	This option is not related to performance. Leave it deselected. (It is not selected by default.)

2. Select the Properties tab and configure according to the descriptions in the following table.

The following table does not describe all properties, but only those that are relevant to performance.

Properties	Description
<code>AlwaysCompressed/NeverCompressed</code>	<p>Allows control of which file extensions and mime types should be compressed. By default all CSS (Cascading Style-Sheets) and JS (JavaScript) files are marked not to be compressed. For performance, we recommend allowing the compression of these files. JS and CSS are text files and generally compress well (~90%). This reduces network traffic significantly.</p> <p> Note</p> <p>The boost in performance is felt significantly over WAN, particularly upon the first (non-cached) navigation actions; in LAN scenarios there is some minor cost in server and browser CPU for the compression/decompression actions. Nevertheless, in almost all cases the advantages outweigh the disadvantages.</p> <p> CAUTION</p> <p>Test content after changing this configuration for CSS and JS files.</p>
<code>CacheControl</code>	<p>Specifies the length of time (in seconds) that resources can stay cached on the browser. The default value is 86400 (24 hours).</p> <p> Recommendation</p> <p>If you do not make frequent changes to your content, consider raising this to one week.</p>
<code>SapCacheControl</code>	<p>Same as <code>CacheControl</code> but applies to saving resources in the Internet Communication Manager (ICM) cache, not the browser. More information: Internet Communication Manager (ICM) on the SAP Help Portal.</p>
<code>MinimumGZipLength</code>	<p>Determines the minimum size of the request to be compressed (in bytes). The benefits of compression in network traffic must be weighed against the increased CPU consumption on the server side for the compression process. The default value is 8KB. Consider lowering the value. A value of 1KB has proven to function well, but trying different values and monitoring the results can help determine the best in any specific landscape.</p>

3. Verify that the configuration is the same for all server nodes.

 **Note**

Any process running on the client machine, such as anti-virus and firewall applications, can affect performance.

2. Using the Navigation Cache to Improve Performance

SAP NetWeaver Portal provides a mechanism for caching navigation nodes to improve performance. The navigation cache is the intermediary between the navigation iViews, such as the Top Level Navigation (TLN) and Detailed Navigation (DTN) areas, or any other content repository. (More information: [Navigation iViews](#) on the SAP Help Portal)

2.1 Background Information

Every portal role comprises a set of navigation nodes that are visible at runtime, upon logon and during navigation in the portal, to the users assigned to that role. The sets of navigation nodes associated with the different roles are determined by navigation connectors. Any number of connectors may be available in a portal.

2.1.1 The Navigation Model

It is helpful to have a minimal understanding of the navigation model that SAP NetWeaver Portal implements, including the existence and function of navigation connectors, to have a more complete idea of the way the navigation cache is configured.

When a navigation event occurs, the following basic flow is initiated:

1. The navigation service is invoked, which implements methods `getInitialNode` and `getNode` to retrieve the navigation nodes relevant to the current user.
2. Depending on the role invoking the navigation service, the relevant navigation connector runs. The navigation connector mediates between the navigation service and the navigation target.
 - o In the case of the portal, the target is the PCD
 - o Navigation connectors can be written for other navigation targets, such as file systems or any database.
3. Before accessing the PCD, the connector first accesses the navigation to check if the current set of navigation nodes has already been requested and cached.

To improve performance, the portal caches each set of navigation nodes required by a user. If another user, having access to the same navigation hierarchy, initiates navigation, the portal can retrieve the navigation hierarchy from the cache instead of generating it again from the PCD.

2.1.1.1 Caching by Navigation Connector

The navigation caching stores nodes in separate areas of the cache for each navigation connector; and for each connector, two node collections are stored:

- **Entry Points:** The nodes returned by `getInitialNodes()`. Entry points are cached with a unique key. For example, for the roles connector, each set of entry points for a given role might be cached with the name of the role as the key.
For each entry point, the cache contains the node name, its hashed value, and its children.
- **Navigation Nodes:** The children of the entry points, and their children.
For each node, the cache contains the key properties of the node, its hashed value, and its children.

This division of the cache into regions, according to connectors and node types, makes cache configuration more selective and efficient, allowing you to configure the navigation cache by entry points or navigation nodes for each connector separately.

2.2 Configuring the Navigation Cache

This section describes the guidelines for how to configure the navigation cache. The feature is disabled by default to avoid the caching of navigation nodes during testing and configuration of the portal. Once the portal is ready for production, caching can be enabled.

Note

Enabling navigation caching for productive systems reduces the number of delta link resolutions, which can be very costly for performance.

The Navigation Cache administration environment permits the following actions:

- Enabling and disabling the cache
- Clearing the cache
- Configuring the cache

In addition to the UI for administrators, the portal provides public APIs for developers.

For API information in the NetWeaver SAP Portal Developer:

[Modifying the Desktop and Navigation](#)

[Navigating in the Portal](#)

2.2.1 Enabling/Disabling the Navigation Cache

In the portal cache management tool, you can enable or disable the entire navigation cache, including all connectors, by pressing *Enable All* or *Disable All*; you can do this also for individual connectors.

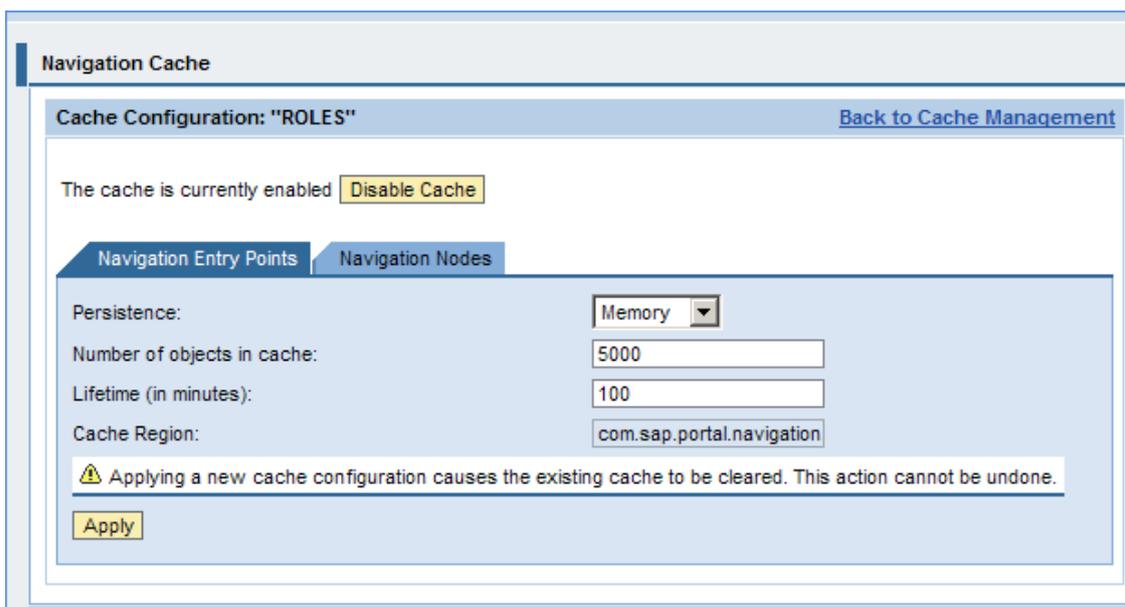
To enable or disable the cache for individual navigation connectors use the following procedure.

1. Access the navigation cache page in the portal by going to *System Administration* → *Navigation* → *Navigation Cache*.

2. In the *Navigation Connectors* table, choose *Configure* in the row containing the connector for which you want to enable or disable caching.

The Cache Configuration page appears, displaying the following information:

- The name of the selected connector in the title bar, within single quotation marks, for example "ROLES"
- An indication of whether or not the cache is enabled followed an *Enable Cache* or *Disable Cache* button
- Caching parameters
- Tabs for configuring the caching parameters separately for navigation entry points and navigation nodes



3. In the relevant tab, choose to enable or disable the navigation cache as required.

2.2.2 Clearing the Connector Cache

If the content administrator changes the navigation hierarchy in any way, for example, by creating a new entry point in a role, the cache is no longer valid until after the cache is cleared or after the cache lifetime is reached.

The following options are available for clearing the cache:

Task	Procedure
Clear the cache for all navigation connectors on all the machines in a clustered environment.	In the <i>Cache Management</i> section of the main <i>Navigation Cache</i> page, choose <i>Clear Cluster Cache</i> .
Clear the cache for a specific navigation connector cluster on all the machines in a clustered environment.	In the <i>Navigation Connectors</i> table, in the row containing the connector that you want to clear, choose <i>Clear Cluster Cache</i> . Note Changing the cache configuration of a navigation connector and clicking <i>Apply</i> also clears the cluster cache for that connector. For more information, see <i>Configure the Cache</i> below.

Clear the cache for all navigation connectors on the local machine only.	In the <i>Cache Management</i> section of the main <i>Navigation Cache</i> page, choose <i>Clear Local Cache</i> .
Clear the cache for a specific navigation connector on the local machine only.	In the <i>Navigation Connectors</i> table, in the row containing the connector that you want to clear, choose <i>Clear Local Cache</i> .

2.2.3 Configuring the Connector Cache

1. In the *Navigation Connectors* table, choose *Configure* in the row containing the connector for which you want to configure cache settings.
2. Configure the settings in the *Entry Points* and *Navigation Nodes* tabs, entering values for the following parameters:
 - *Persistence* – the persistence level for which you are configuring the connector cache
 - Memory
 - Database

More information: [Determining the Cache Persistence Mode](#) on page 26

- *Number of objects in cache*
- *Lifetime (in minutes)*

For guidelines about determining the number of objects in the cache and setting the cache lifetime: [Tuning the Connector Cache Configuration](#) on page 22

- *The Cache Region parameter is read only.*

Recommendation

The configuration settings in the *Entry Points* and *Navigation Nodes* tabs do not have to be identical, but it is recommended not to specify different *Persistence* settings.

3. Click *Apply* to save the new configuration.

Important

Changing the navigation cache configuration clears the entire cache. Once applied, this cannot be undone.

2.2.4 Tuning the Connector Cache Configuration

Following are guidelines and recommendations for setting the properties for number of cached objects and cache lifetime according your requirements:

- Estimate the number of objects expected in the cache.

The way to do this is to log on with a user who belongs to the required, representative roles relevant to your organization and then view the cache contents (more information: [Viewing Cache Content](#) on page 23).

It may be helpful to use the service `CachePreloaderService`, which enables you to upload all the roles of a specified user into the navigation cache (more information: [Preloading Roles](#) on page 24).

More information: [Configuring the Navigation Cache](#) on the SAP Help Portal

Recommendation

If X is the expected number of objects in cache, define the maximum cache size as $100X / 70$; in other words, the recommended value for the expected number of cached objects is 70% of the maximum cache.

- View the contents of the navigation cache frequently to maintain awareness of significant changes to the quantity of cached objects.
- *Lifetime* refers to the time, in minutes, that cached objects are valid.

Determining a value for this property requires being aware of how often navigation content is changed and defining the cache lifetime accordingly. The default lifetime is 100 minutes.

2.2.5 Viewing Cache Content

Each cache connector comprises two cache regions: one for initial nodes, or navigation entry points, the other for all nodes, which are not entry points. You can view either the entry points or other nodes by opening the respective tabs.

In the *Cache Content* screen, you can display a list of all the objects currently in the cache. View the contents as follows:

1. Click the *Show Contents* link of any connector to reach the Cache Content page for that connector.
2. Using regular expressions, specify search criteria to find the objects that you want to view.

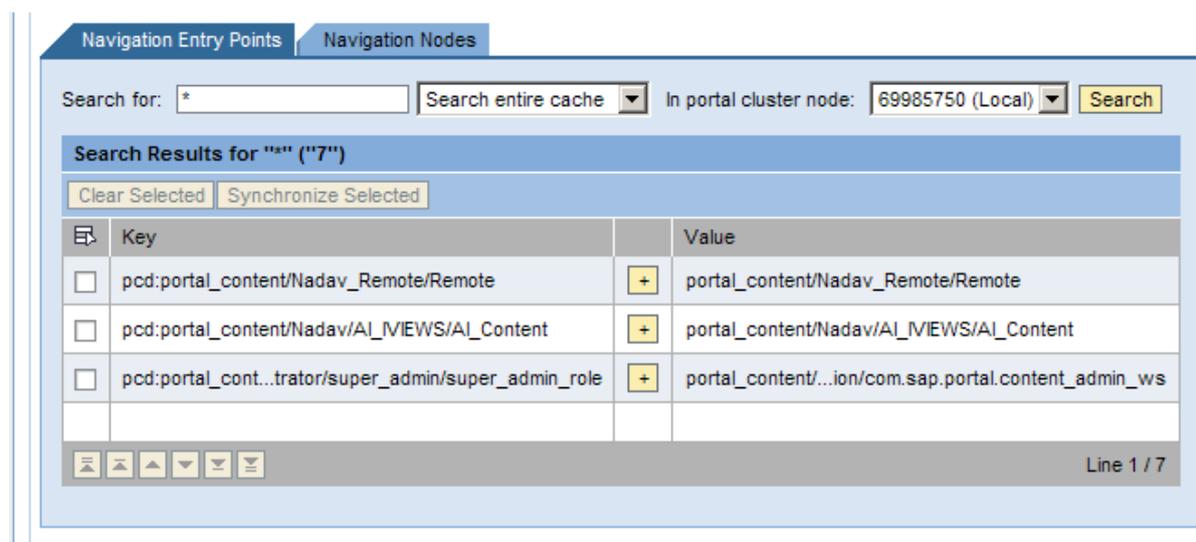
You can search by key, by value, or by part of the node URL, for example, **myPage**. The search mechanism finds all of the expressions that appear in the cache entry value, including all of the navigation attributes that may be displayed there, such as the node title, launch URL, show type, and description.

You can search the entire cache or limit the search to the results already displayed in the table. You can also limit the search to a specific cluster node.

3. Click *Search*.

The results table displays the objects that match the specified search criteria. The display limit is 5000 objects.

- To display the details of an object in the results table, click the plus sign (+) next to the object.
- To clear specific objects from the cache, select the checkbox next to the object and click *Clear Selected*.



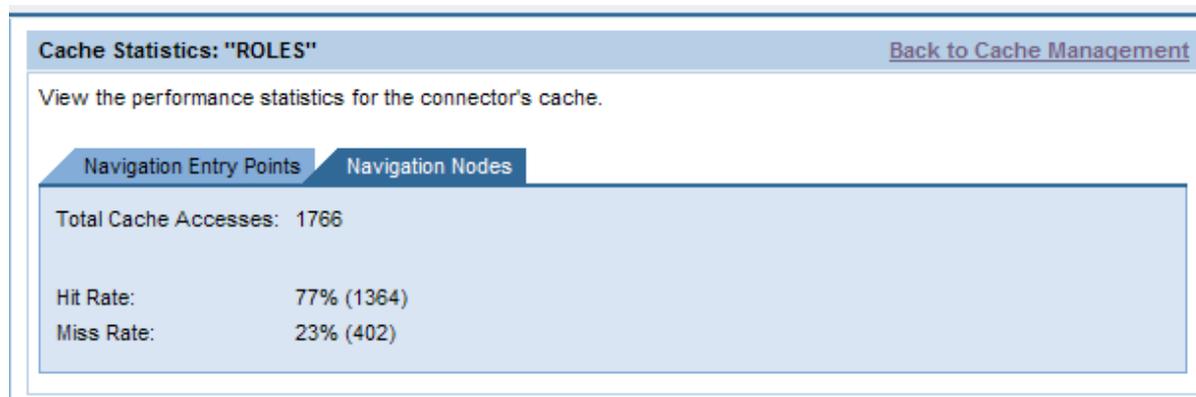
The screenshot shows the 'Cache Content' interface with two tabs: 'Navigation Entry Points' and 'Navigation Nodes'. The search criteria are set to '*/7' and 'Search entire cache'. The results table displays three entries:

Key	Value
<input type="checkbox"/> pcd:portal_content/Nadav_Remote/Remote	portal_content/Nadav_Remote/Remote
<input type="checkbox"/> pcd:portal_content/Nadav/AI_VIEWS/AI_Content	portal_content/Nadav/AI_VIEWS/AI_Content
<input type="checkbox"/> pcd:portal_cont...trator/super_admin/super_admin_role	portal_content/...ion/com.sap.portal.content_admin_ws

At the bottom right of the table, it indicates 'Line 1 / 7'.

2.2.6 Displaying Cache Statistics

You can view the performance statistics for the cache of each connector by clicking the *Statistics* link in connector row of the *Navigation Connectors* table. The link returns the total number of cache hits and the hit and miss rates.



Cache Statistics: "ROLES" [Back to Cache Management](#)

View the performance statistics for the connector's cache.

Navigation Entry Points | Navigation Nodes

Total Cache Accesses: 1766

Hit Rate: 77% (1364)

Miss Rate: 23% (402)

2.2.7 Additional Cache Services

The following services give you additional capabilities when administering the navigation cache:

- CacheCleanerService
- CachePreloader

2.2.7.1 Clearing the Cache at Set Intervals

You set the `CacheCleanerService` to automatically clear the navigation cache at specified intervals. You can clear the cache for all connectors or for specific connectors.

1. Go to *System Administration* → *System Configuration* → *Service Configuration*.
2. In the Portal Catalog, open *Applications* → `com.sap.portal.navigation.service` → *Services*.
3. Right-click `CacheCleanerService` and choose *Configure*.
4. Set the following properties:
 - `ClearCacheIntervallInDays`: Set the interval after which to clear the cache.
Default value: 0
 - `ListOfConnectors`: List the connectors that you want to clear. If empty, all connectors are cleared. Separate entries with a semicolon (;).
Default value: ROLES

By default, this service starts together with the portal. However, because the default interval is 0, unless you set a new interval, the connector caches are not cleared.

2.2.7.2 Preloading Roles

The `CachePreloaderService` allows you to upload all of the roles of a specified user into the navigation cache. Both roles and short URLs are uploaded.

Preloading for SAP NetWeaver 7.0 Including SAP Enhancement Package 1

1. Go to *System Administration* → *System Configuration* → *Service Configuration*.
2. In the Portal Catalog, open *Applications* → `com.sap.portal.navigation.cachepreloader` → *Services*.
3. Right-click `CachePreloaderService` and choose *Configure*.
4. Set the following properties:
 - *Run on startup*: Specify whether or not the preloading of roles is performed when the service starts.
Default value: `false`
 - *Users to Preload*: List the users whose roles you want to preload, in the following format:
`[<user name>:<number of navigation node levels to preload>:false:<locale1>,<locale2>]`
For example: `[Administrator:2:false:en,de]`
Separate entries with a semicolon (;).

If the cache is cleared, you need to restart the service.

Important

Make sure that you restart services after making changes.

Preloading from SAP Enhancement Package 2 for SAP NetWeaver 7.0

From this release, the `CachePreloaderService` service is deprecated. However, you can configure the same settings for preloading roles in the navigation cache interface in the portal.

1. Navigate to *System Administration* → *Navigation* → *Navigation Cache*.
2. In the *Cache Preloader Service* section, choose *Configure*.
3. Define when to run the cache preloader service:
 - Run the service every time the cache is cleared.
 - Run the service at specific intervals.

You can choose both checkboxes so that the service runs at specific intervals as well as running every time the cache is cleared.

Note

You can also run the service at any time by choosing *Run Cache Preloader Service* in the *Cache Preloader Service* section of the main cache management screen.

4. Specify the users roles to upload. In the *User Details* section, enter the following details:
 - a. *User ID*: Enter the ID of the user whose roles you want to upload and choose *Validate*.
The default language of the user appears in the *Languages* field.
 - b. *Additional Languages*: Add additional languages for the user.
 - c. *Navigation Levels*: Specify the number of navigation levels that you want to upload with the role.
5. Choose *Save User*.
The user appears in the *List of Users* section.
6. After you have added users, you can edit their details or remove them from the list.

7. Choose *Save Configuration*.

2.2.8 Determining the Cache Persistence Mode

Upon configuring your connector setting, there is an option to set the cache persistence mode to be in memory or in the database. By default the cache persistence mode is set to Memory.

Important

If a customer needs cache cluster synchronization, meaning same cache content should be in all cluster nodes then the cache must be configured to use database persistency.

If the cache persistence mode is set In Memory, the cache on each cluster node contains the content that was loaded on it by demand; the result is that different cache content is loaded in each cluster node.

2.2.9 Using Traces to Check the Cache

You can use the information obtainable from the navigation cache traces to view the activity in the cache and draw conclusions about the need for fine tuning.

- location: `com.sap.portal.navigation.cache`
- Severity: `DEBUG`

Typically, common attributes, such as `title` and `launch URL`, are loaded to cache when the navigation node loads. The cache log traces indicate, at runtime, which navigation attributes and navigation nodes were taken from cache and which required loading from the connector repository, which for the `ROLES` connector is the PCD.

A connector developer can use the traces to find which attributes require additional access to the PCD, or other repository. On the basis of the trace information, it can be determined how to optimize calls to the connector repository by identifying which common attributes to load to cache.

The portal provides APIs to set a profile, for each connector, that can define additional connector-specific attributes to load to cache when the connector node is saved.

The public API for this is `setConnectorNodeCachedAttributes` in class `NavigationConnectorProfile`.

For more information about navigation, see the SAP NetWeaver Developer's Guide and go to the section *Running an Enterprise Portal → Go and Create → Modifying the Desktop and Navigation → [Navigating in the Portal](#)*.

3. Configuring Portal Runtime for Performance

Portal Runtime (PRT) provides a Java-based framework for running applications that display content in the portal. You can configure properties that affect performance in places:

- In properties of the J2EE service `com.sap.portal.prt.sapj2ee`, using the J2EE Visual Admin tool
- In the portal runtime central configuration in the portal administration environment

3.1.1 Configuring the J2EE Portal Runtime Service

1. Open the Visual Admin tool on the portal server by navigating to `...\\usr\sap\E16\JC90\j2ee\admin\` and running `go.bat`.
2. Navigate to *Server* → *Services* of the service `com.sap.portal.prt.sapj2ee`.
3. Modify the properties that are relevant to your organization.
4. Save and restart the service.

Property	Default Value	Description
<code>EnableLogging</code>	<code>false</code>	<p>Enables logging of information messages to the <code>com.sap.portal.sapj2ee.infoWarning</code> location.</p> <p>To enable logging, also configure also the <code>infoWarning</code> location to <code>info</code> or <code>ALL</code>.</p>
<code>async.response.pool.size</code>	<code>100</code>	<p>The thread pool size for the asynchronous response mechanism</p> <p>This value depends on an estimation of the number of embedded iViews per page.</p> <p> Recommendation Recommended value is <code><engine pool size> * 2.5</code></p> <p> Note The calculation of this default value is based on an engine pool size of 40 concurrent users multiplied by an estimated number of 2.5 embedded iViews per page.</p>
<code>async.response.timeout.pool.size</code>	<code>100</code>	<p>The thread pool size for managing the timeout for when the asynchronous response mechanism is used.</p> <p> Recommendation Recommended value is the same as <code>async.response.pool.size</code>.</p>
<code>caching.persistence.clean.period</code>	<code>5</code>	<p>The interval, in minutes, for cleaning the database cache.</p>

<code> caching.persistency.infinite.validity.clean.period.factor </code>	144	<p>One of the parameters that determine the validity period for items marked as having infinite validity.</p> <p> Important</p> <p>Entries with infinite validity might overload the database.</p> <p>The validity period is the value for this property multiplied by the value for the property <code> caching.persistency.clean.period </code>.</p> <p>The default value results from $144 * 5 = 12$ (hours)</p>
<code> caching.maxsize </code>	5000	The maximum size of the cache, in kilobytes, before clearing the cache to make room for new items.
<code> pooling.string.buffer.maxsizebytes </code>	20000000	<p>The maximum size, in bytes, for each virtual machine (VM) for string buffer pools.</p> <p>Special values:</p> <ol style="list-style-type: none"> 0: No pool -1: Infinite pool

3.1.2 Configuring the Portal Properties

1. In the portal, navigate to *System Administration* → *System Configuration* → *Service Configuration*.
2. In the Portal Catalog, expand the *Portal Runtime* folder.
3. Right-click *Central Configuration* and in the context menu, choose *Configure*.
The PRT central configuration page appears.
4. Edit the properties as necessary and choose *Save*.

Property	Default Value	Description
<code> caching.cluster.notification </code>	true	<p>Enables the use of the notification service to invalidate cached objects in a cluster configuration.</p> <p>Set this property to true when the <code> caching.persistency </code> property is set to true.</p> <p>Setting the value to false is useful only for problem isolation when there is a high number of notifications caused by cache invalidation.</p>
<code> caching.off </code>	false	Enables portal caching

<code> caching.persistence </code>	true	<p>Enables persistency caching for all caches</p> <p>Set this property to false for isolating database problems.</p>
<code> enable.http.conditional.request </code>	true	<p>Enables the handling of HTTP conditional requests by the cache.</p>
<code> Portal.runtime.exception.printcause </code>	false	<p>Prevents excessive printing of exceptions when set to false.</p> <p>More information: SAP Note 937737</p>
<code> caching.portalcomponent.persistence </code>	true	<p>Enables persistency caching for the portal component cache.</p> <p> Important</p> <p>The property <code> caching.persistence </code> must also be true.</p> <p>Set this property to false for isolating database problems.</p>
<code> jsp.bigmode.delimit.size </code>	10000	<p>Specifies the use of code optimization for JSP files larger than the specified number of bytes. This enables the extension of the standard byte code limits used by the Java standard.</p> <p>More information: SAP Note 820282</p>

4. Configuring the Portal Content Directory (PCD) for Performance

The Portal Content Directory (PCD) is the main repository for portal content, whether delivered with the portal, created by administrators, or developed and deployed by portal content developers. The PCD contains a hierarchy of folders, each of which can contain semantic objects, such as iViews, pages, and roles.

Important

Developers of portal content need to pay attention to the performance aspects of the application they create. More information: [Developing Well Performing Portal Applications](#) in the SAP NetWeaver Portal Developer's Guide..

The purpose of this section is to inform system administrators of the PCD configuration settings that can improve portal performance.

4.1 Overview of the PCD

This section describes the structure and terminology of the PCD. The PCD is stored in the portal database and is accessed through the PCD Generic Layer (GL), which is a hierarchy-based, JNDI provider, extended to implement the following additional functionality:

- Personalization: The PCD enables portal objects to be personalized. For each attribute of each portal object, different values can be stored for each user.
- Delta links: The PCD enables the creation of portal objects whose attributes are inherited from another portal object. Changes to the original object update the delta links.

A delta link may be seen as similar to inheritance in object orientation. Modifications to the source affect the delta link but modifications to the delta link do not affect the source. All kinds of modifications are supported: attribute changes, creation, deletion, resorting, creation of delta links.

Each object in the PCD is referred to by its PCD name, which is the full path to the object through the PCD tree structure. The object is referenced within the system by its URL, which begins with the protocol `pcd:`, for example, `pcd:portal_content/myFolder/stocks`.

This points to a PCD object called `stocks`, in a folder called `myFolder`, under the top-level folder `portal_content`. The atomic name of a PCD object is the name of the object, without the entire path. In the example above, the name `stocks` is the atomic name of the object; this name must be unique within the folder that contains the object.

More information: [Portal Content Directory \(PCD\)](#)

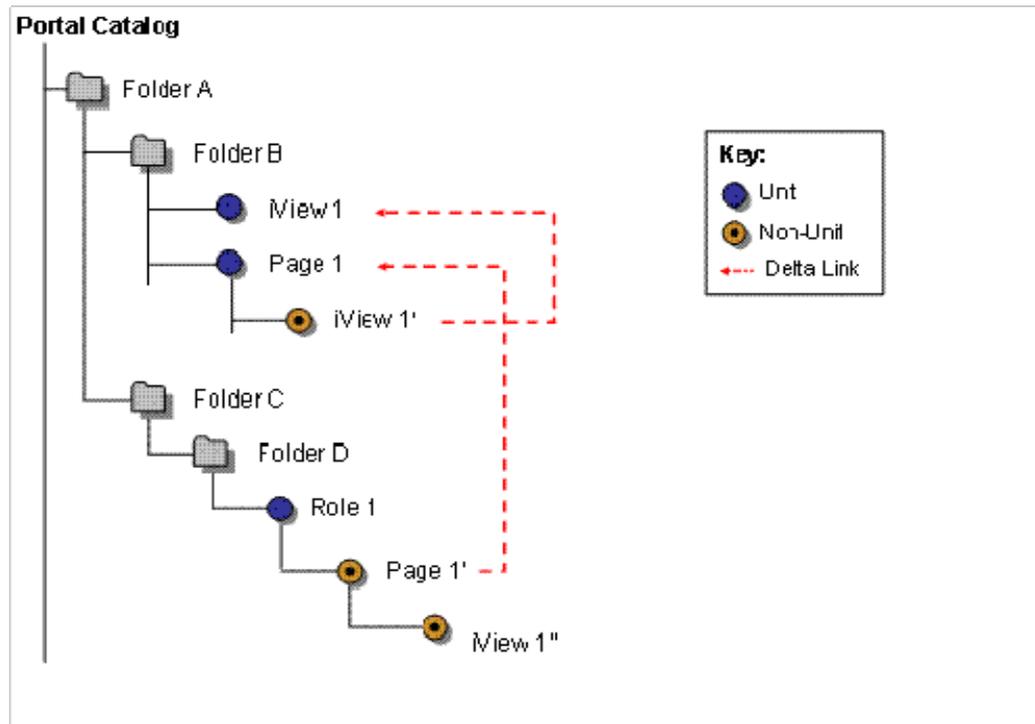
4.1.1 PCD Structure

PCD content is organized into a hierarchy of folders and subfolders, some of which are out-of-the-box; administrators can add folders to the hierarchy.

4.1.2 Unit Objects

The contents of folders, which are not folders themselves, are unit objects. Unit objects are semantic objects, such as iViews, pages, and roles, whose parents in the PCD hierarchy can consist only of plain folders. (A plain folder has no other parent hierarchy than other folders; a folder within a role is not a plain folder.) A unit object itself may also include a complete child hierarchy, such as a page

containing pages and iViews, or a role with associated content. All children of unit objects are subunits.



The following list describes some of the many tasks that can be performed on unit objects only:

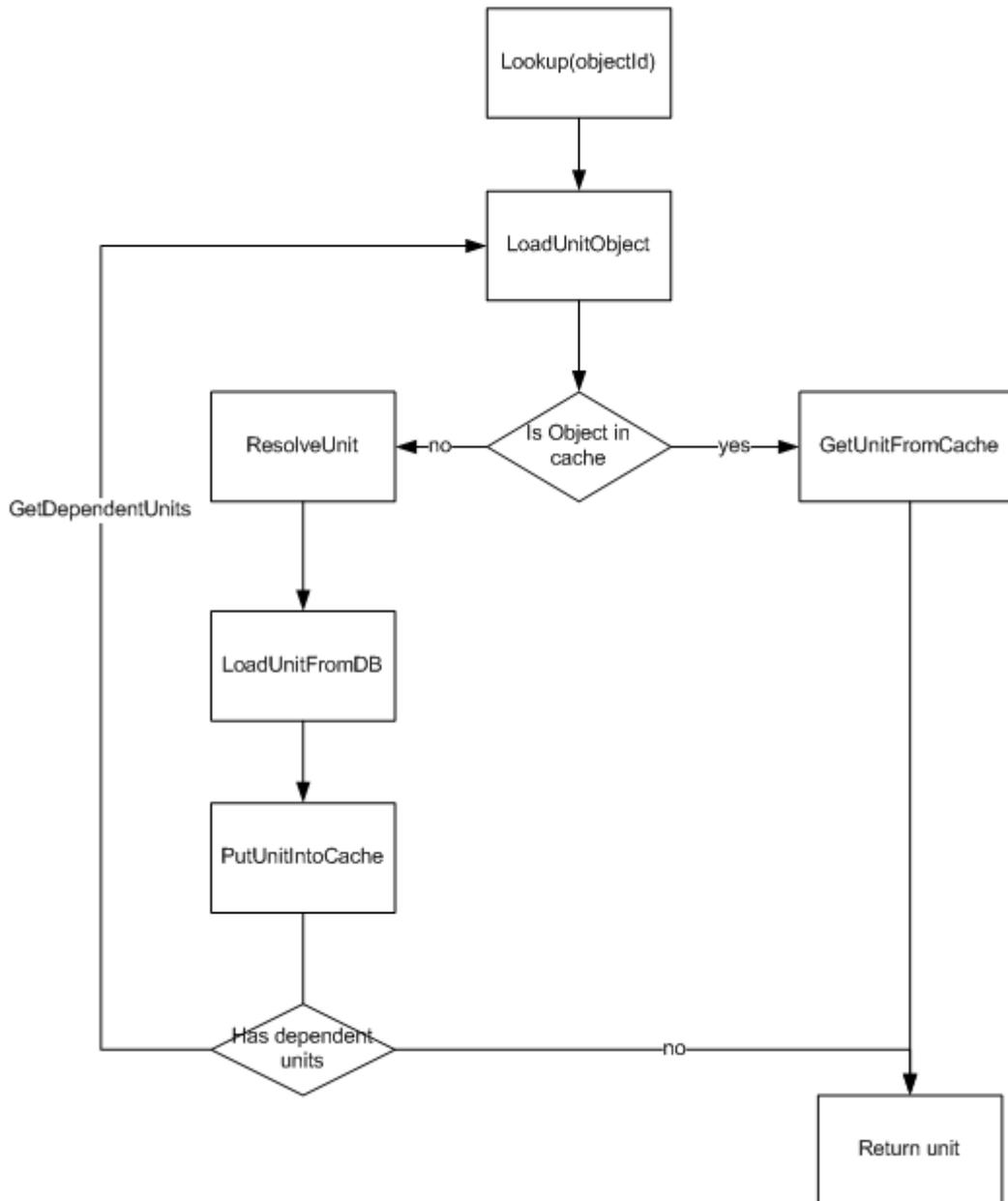
- Delta Link creation: Only unit objects can be targets of delta links.
- Transport: Only unit objects can be transported.
- Translation: Translation worklists include only unit objects.
- Caching: Only unit objects can be cached.

4.1.3 Loading Time of a PCD Object

Loading time of a PCD object, when it is not in the cache, is mainly affected by following factors:

- Number of delta-links used, affecting the number round trips to the database
- Number of attributes, affecting the size of the query result
- Number of objects directly contained, affecting the size of query result

The following illustration shows the loading sequence of an object:



4.1.4 Factors Affecting Performance

Units are retrieved separately from the persistence layer and delta links point to new units that are loaded separately when information is accessed. To retrieve information from a delta link, all target objects have to be loaded completely (to apply the delta).

Loading time of a PCD object, when it is not in the cache, is mainly affected by following factors:

- Number of delta-links used, affecting the number round trips to the database
- Number of attributes, affecting the size of the query result
- Number of objects directly contained, affecting the size of query result

- Number of shared objects – The more that objects are shared, the higher the probability that the objects are retrieved from the PCD cache
- Amount and type of personalization – Less personalization benefits performance, especially personalization that involves structural changes (adding or deleting of objects); attribute changes do not significantly affect performance
- Optimization of PCD (EP) database tables, for example, indexing and cleaning
- Number of changes in delta link chains, especially structural changes

4.2 General Guidelines and Tips

You can improve portal performance by understanding how the PCD functions and by adopting the principles for managing content described in this section.

Delta-link resolution in the PCD necessitates the loading of the entire subtree from the current node on the portal server. This means that if an entry point is located within a delta-linked object, such as a role, all related objects (delta links) are loaded, building a virtual tree. Virtual trees are built on demand but they are always built completely and therefore decrease performance.

Recommendations:

Configure the VM parameter `-XX:SoftRefLRUPolicyMSPerMB=1`, as recommended in SAP Note [723909](#)

More information: [Background](#)

- Optimize object size

Run load tests to optimize object size so that as many PCD objects as possible are in the cache.

Large units having a complex structure also have a higher memory footprint, which causes quicker removal from the cache. Often only a small part of the data is really needed.
- Avoid the use of delta links to purely navigational, hierarchical objects (worksets and roles).

 **Tip**

Of course if these objects are often reused, consider using delta links instead of rebuilding the entire structure.
- Use delta-links for commonly used objects
- Keep delta link chains as short as possible
- At runtime, attributes are shared in memory. Making changes to object attributes only on source objects saves memory.
- Avoid frequent updates to central objects

Notification may be slow if objects are referenced by a large number of delta-links and updates are made often.
- Avoid nesting large objects, for example, worksets within worksets within worksets
- Reuse smaller units frequently for better cache utilization; and optimize iView reuse by using generic iViews (for example, BW iViews)
 - The use of many small units increases the number of database round trips, but has a lower memory footprint. Small generic objects load quickly into cache.
 - High reuse results in better PCD cache efficiency because a lower number of distinct objects need to be loaded.

- Use only one entry point for the larger roles
This avoids the unnecessary loading of objects that are outside of the current navigation path.
- Use delta links cautiously, especially for roles and worksets
- If an entry point within a role is not reused, use a folder inside the role.

4.3 Configuring PCD Properties

This section presents the PCD properties that affect portal performance. (More information at help.sap.com: [Configuration for the Portal Content Directory](#) in the SAP Help Portal).

The property descriptions in the following table relate to the cache parameters of the persistence layer of the portal:

PL Cache Parameters	Description
<code>Pcd.Pl.ShortStringCacheSize</code>	<p>The maximum number of converted strings.</p> <p>Long strings are converted into short strings, for improving performance, and then reconverted back to the original long strings.</p> <p>Default value: 1000</p>
<code>Pcd.Pl.AttrIdCacheSize</code>	<p>Cache size for attribute IDs that are shared between different objects.</p> <p>Default value: 5000</p>
<code>Pcd.Pl.MaxLengthSharedAttributeValueString</code>	<p>String sharing does not apply for strings that exceed this maximum length.</p> <p>Default value: 50</p>
SoftCache Service Parameters	Description
<code>Pcd.SoftCache.HardReferenceLimit</code>	<p>The number of objects that the PCD holds in memory as hard references.</p> <p>Default value: 1000</p> <p>Note that this property is generally superseded by the PCD cache optimizer property <code>Pcd.Xfs.Cache.HardReferenceLimit</code></p> <p>This property may not be in use.</p>
<code>Pcd.Texts.UseOriginalLocaleOnly</code>	<p>Boolean Parameter: When The Value Is <code>true</code>, is useful in a single-language system, saving one round trip to the database.</p> <p>Default value: <code>false</code></p>

4.4 Optimizing the PCD Cache

The cache efficiency of the Portal Content Directory (PCD) can be fine-tuned to optimize system performance. The object is a configuration that retains a maximum number of cached objects within the limits of available memory.

4.4.1 Background

The PCD caches objects that are loaded from the database. The cache is designed to be memory sensitive: frequently used objects that are currently used by client applications, or likely to be reused soon, are kept in the cache as long as there is sufficient memory.

The Java class `java.lang.ref.SoftReference` implements this behavior. Depending on the Java VM runtime implementation and system load, the use of soft references may require an aggressive soft reference garbage collection policy to ensure stable runtime behavior. This means that soft reference objects are cleared quickly. (SAP Note [723909](#) contains current recommendations.) However, an aggressive soft reference policy can significantly reduce the efficiency of the PCD cache.

A high eviction rate of frequently used objects can be reduced by tuning the PCD cache optimizer parameters. Hard references are used for certain cached objects and so avoids the eviction of these objects due to soft reference clearing.

If the cache optimizer is activated, it stores access statistics for the cached objects. The optimizer thread periodically checks the cache and changes the status of the most frequently used objects from soft to hard-referenced cache entries. An optimal configuration keeps all objects in the cache that would, if cleared, soon be reloaded; and allows the clearing of less frequently used objects to free memory.

4.4.2 PCD Cache Configuration Parameters

This section describes the recommended configuration for PCD cache optimizer parameters. For information about changing PCD configuration parameters, see [Configuration for the Portal Content Directory](#) on the SAP Help Portal.

PCD Cache Optimizer Parameters	Description
<code>Pcd.Xfs.Cache.StartCacheOptimizer</code>	Flag to activate/deactivate the cache optimizer. Default value: true
<code>Pcd.SoftCache.OptimizerTickLength</code>	Time in milliseconds that determines the frequency of cache optimizer traversal runs. Default value: 300,000
<code>Pcd.SoftCache.OptimizerWaitAfterVmStartup</code>	Time in milliseconds for the delay of starting the cache optimizer during system startup. Default value: 300,000
<code>Pcd.Xfs.Cache.HardReferenceLimit</code>	Number of cache entries referenced by hard references. Default value: 100 If the value is too low, the number of database requests increases; if the number is too high, memory consumption increases.
<code>Pcd.Xfs.Cache.ObjectClassLevel1</code>	List of <i>objectclass</i> names (separated by a comma ";" or semicolon ";" character) that are hard-referenced with medium weight

`Pcd.Xfs.Cache.ObjectClassLevel2`

List of object class names (separated by a comma ";" or semicolon ";" character) that are hard-referenced with high weight

In the event of memory problems, It is recommended to consider assigning level class 2 to folders.

4.4.3 Recommended Configuration Settings for Cache Optimizer

The default installation settings for cache sizes are configured to fit for low-end system configurations, such as development systems on 32 bit platforms. For 64 bit platforms, it is strongly recommend to increase the hard-reference cache size. The parameter `Pcd.Xfs.Cache.HardReferenceLimit` should be set to at least 500.

Fine Tuning the Cache Optimizer

The goal of fine tuning is to configure the cache optimizer so that the cache hit ratio is as high as possible, while the memory consumption of the cached objects remains within the available boundaries recommended for the PCD.

The metrics used for fine tuning are:

- Overall response times for portal iViews
- Load to PCD database tables (number of database calls to tables starting with `EP_`)
- Memory consumption

One option for monitoring the first two metrics is using SAP Solution Manager Diagnostics. Memory consumption is monitored best by analyzing the verbose output of the garbage collector. Before starting any configuration changes, a baseline measurement should be made for all metrics.

More information about the SAP Solution Manager: [Monitoring \(Overview\)](#) on SAP Help Portal

Relevant Trace Level and Logging Information

To obtain information about PCD cache behavior, the following trace location should be activated:
`com.sap.portal.pcd.Gl.Xfs.Cache.Optimizer = INFO`

Using this trace level, the following information is logged:

- Each time an object is loaded from the database, a line is logged, such as:
`Persistence access for <object-name>`
- At the end of each traversal of the cache by the optimizer thread, the following information is logged:
`[CacheManager.traverse()] Optimize ...`

The following table contains descriptions of information written to the log:

Parameter	Description
TIME	Elapsed time in milliseconds required by the cache optimizer for the traversal of the cache
OBJECT COUNT	Total number of cached PCD objects

HARD_REF	Number of hard-referenced PCD objects
SOFT_REF	Number of soft referenced PCD objects
HARD TO SOFT	Number of cache entries for which references were changed from hard to soft
SOFT TO HARD	Number of cache entries for which references were changed from soft to hard
XFS FROM PERSISTENCE	Number of objects that have been read from persistence since the last cache optimizer traversal
USED MEMORY	Total amount of memory currently used by the Java VM (currently used heap size) in bytes
FREE MEMORY	Amount of free heap memory of the Java VM in bytes
TOTAL MEMORY	Total available memory of the Java VM (maximum heap size) in bytes

During the traversal of the cache optimizer thread, a line for each reference type setting is logged as follows:

```
[CacheManager.keepStrongReference()] set HARD_REF: <object-name>
```

or

```
[CacheManager.keepSoftReferenceOnly()] set SOFT_REF: <object-name>
```

If the reference settings for the same objects oscillate, that is, switch frequently from hard to soft references and back again, during subsequent traversals, it is an indication that the configuration is not optimal. This behavior indicates that these objects should always be cached by hard references because they are used frequently and likely be reloaded shortly after eviction.

Another hint for suboptimal cache behavior is a high number of database round trips. If this is the case, the `Pcd.Xfs.Cache.HardReferenceLimit` should be increased.

Recommendation

Determine the usual size of the cache by estimating a median `OBJECT_COUNT` value from several runs. Typically, the hard reference limit should vary between 1/3 and 2/3 of this value, depending on the amount of memory used by the hard-referenced objects.

Be sure to tune the system when it is running under normal user load for an adequate time (at least 1 day). After any configuration change, the performance metrics and the garbage collection behavior should be re-analyzed.

CAUTION

Too high a setting for the number of hard referenced PCD objects (`Pcd.Xfs.Cache.HardReferenceLimit`) can cause out of memory exceptions.

Usually it is enough to configure the `HardReferenceLimit` parameter to gain good cache efficiency and performance. By default, the PCD cache optimizer considers only the number of runtime accesses to an object as selection criteria for hard-referenced objects. Additionally, the PCD cache optimizer allows differentiation in the importance of PCD objects according to their type; this means that it is possible to define a list of object classes to be cached with higher importance. Technically, the number of cache accesses for objects with object classes defined by the parameters

Pcd.Xfs.Cache.ObjectClassLevel1 and Pcd.Xfs.Cache.ObjectClassLevel2 are multiplied with a weight factor, so that their relevance is increased.

4.5 Benchmarks

Important

DISCLAIMER: A multitude of factors influence performance; to list just a few: the number of roles assigned per user, the number users that share the same role, the number of delta links, cache configuration, system utilization in terms of CPU and memory, scenario.

The purpose of a benchmark is to provide an estimation of expected results in a specific landscape with a specific scenario, described below, in this case, the scenario given in this section.

Note

Benchmark measurements were taken using LoadRunner.

4.5.1 Fixed Test Parameters

The benchmark scenario ran with the following parameters:

- Hardware
 - 64Bit G1 machines (2GHz X 4CPUs, 8GB RAM) with the database on a remote machine
 - Two server nodes, each server with 2 Giga heap size.
 - Other VM settings according the note 723990
- Portal version – NW04s 700 SP18
- Operation system – Windows 2003 Enterprise Edition SP2

The scenario itself comprises the following:

- Logging on
- Two navigation actions to PRT content pages, each page containing 4 iViews
- Three navigation actions to Web Dynpro pages
- Logging off
- The deployed content and used consists of real ERP business package content, including 147 roles.
- The number of rows in PCD tables were as follows:
 - EP_OBJECTS 96,603
 - EP_ATTR_HEADERS 819,543
 - EP_ATTR_VALUES 391,351
 - EP_ATTRS1 391,351
- There were 50,000 named users in the database
- The navigation cache enabled
- The PCD cache optimizer disabled
- Short URL disabled
- DL depth was fixed to the structure of the ERP business package

- Test duration was 5 hours with 800 concurrent users; TPS (transactions per sec) =21

4.5.2 Purpose

The purpose of the benchmark is to check role scalability and the effect of an increased number of roles assigned per user.

The benchmark checks the influence on performance when the number of roles per user increases from 10 to 50 to 100, assigned with a fixed role-sharing ratio of 50% (5 shared, 25 shared and 50 shared respectively).

The roles that are not shared are divided among two user groups for the tests that encompass 10 and 50 roles per user; and between two user groups for the test that encompasses 100 roles per user.

4.5.3 Results

Response Time

The results of the benchmark show that the average response time is three times higher when comparing 10 roles per user to 50 roles assigned per user. The response time is almost doubled when comparing 50 roles per user to 100 roles per user.

10 roles per user: 0.26 seconds

50 roles per user: 0.775 seconds

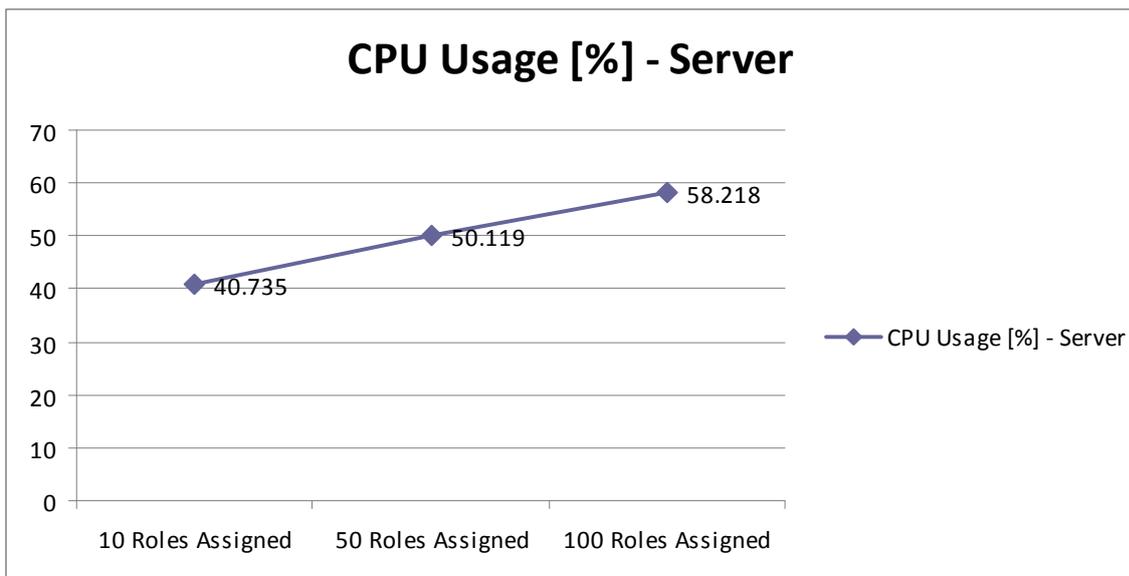
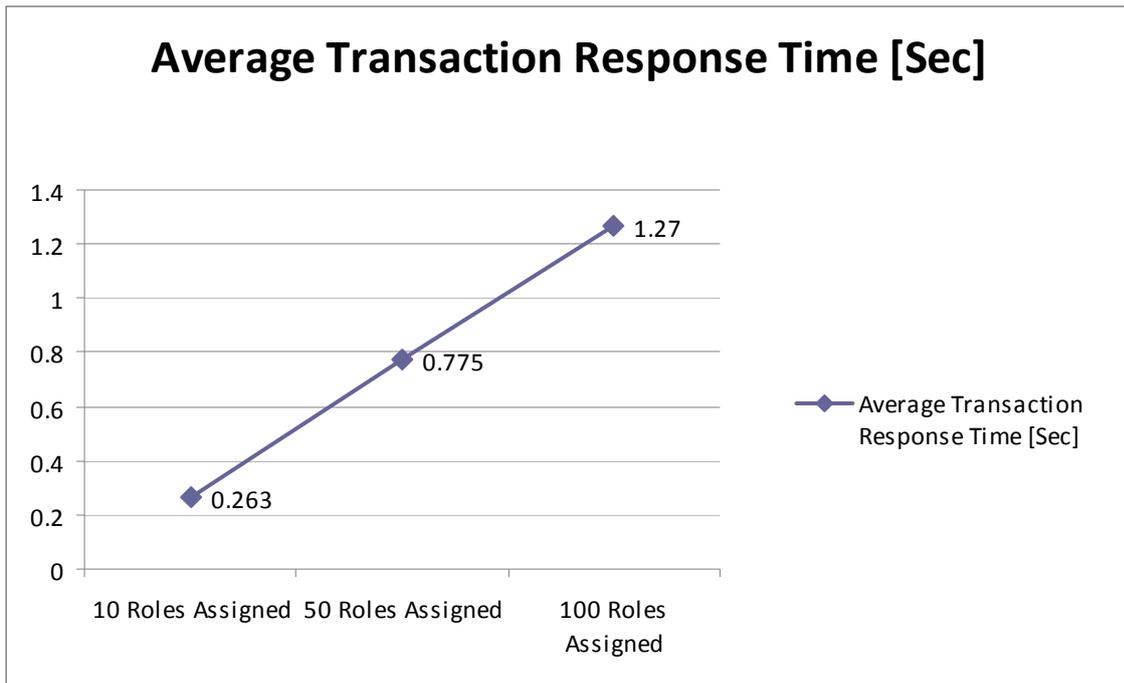
100 roles per user: 1.27 seconds

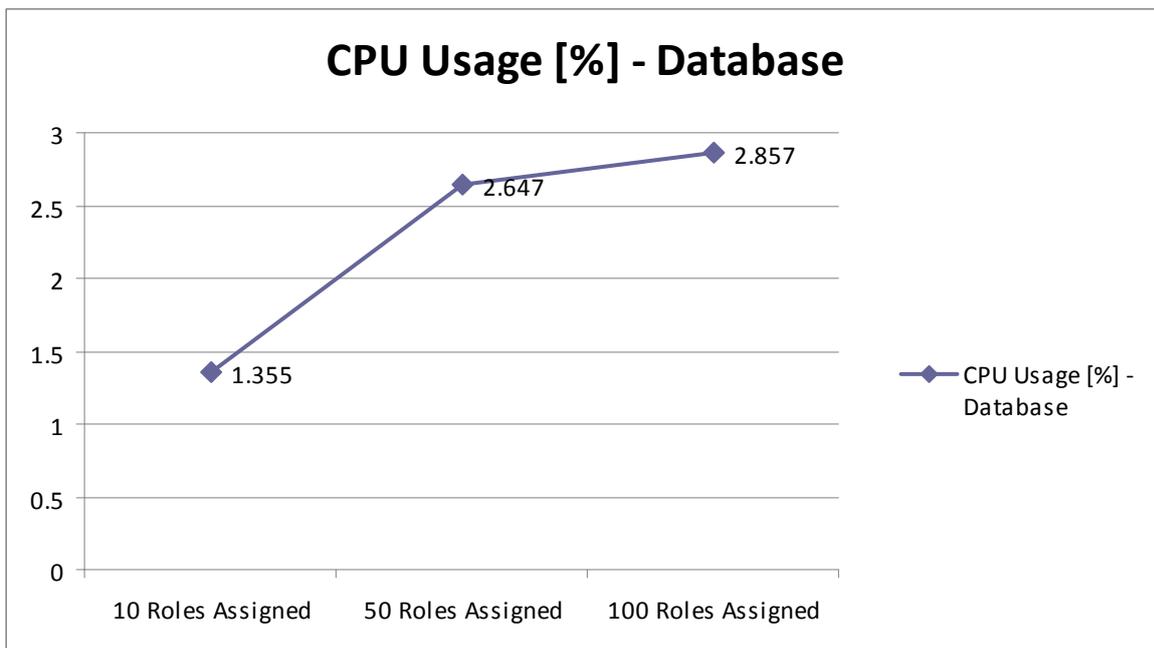
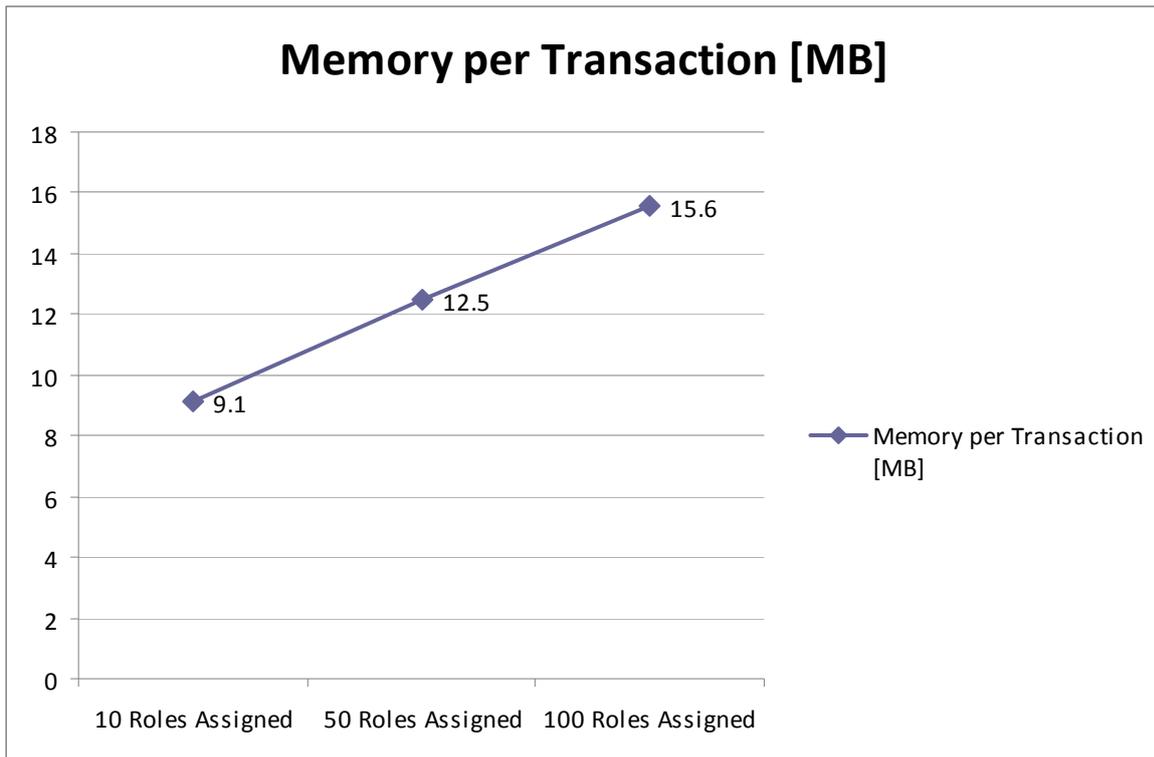
CPU Consumption

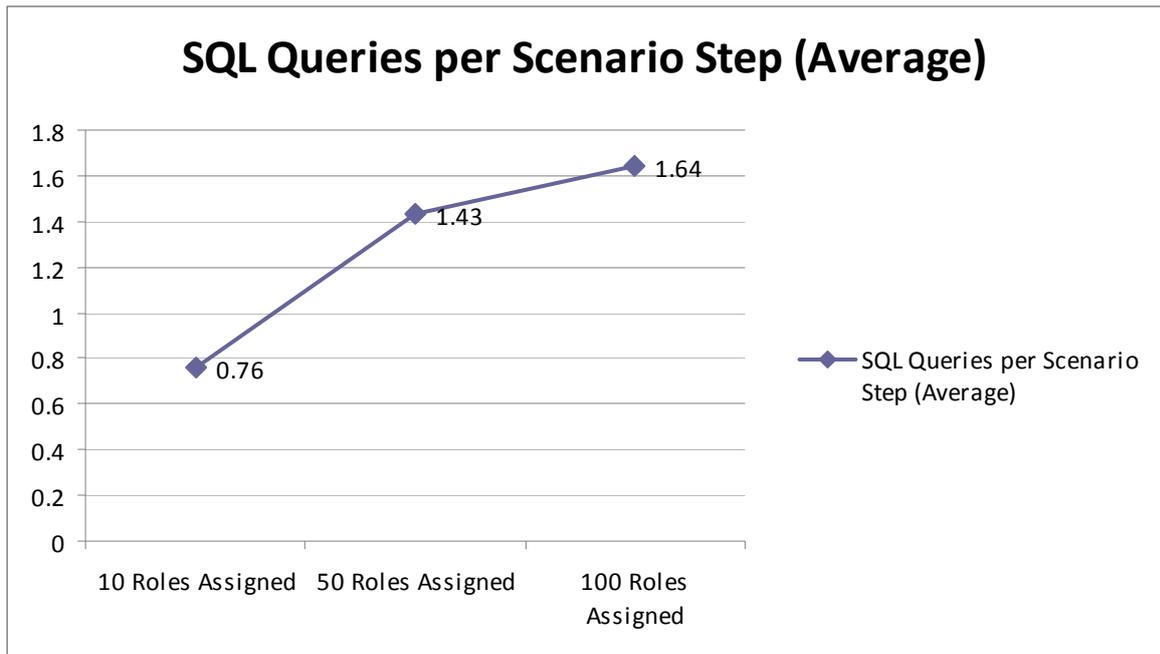
The server CPU use grows from 40% to 50%, comparing 10 roles per user to 50 roles per user and from 50% to 58% comparing 50 roles per user to 100 roles per user.

Conclusion

The number of roles assigned to a user affects memory consumption, CPU consumption, database activity, and response times.







Effect of Role Sharing

The purpose of the following benchmark is to check how performance is affected by role sharing, that is, multiple users having the same role. We used 50 as the fixed number of roles per user.

The variable against which the results were checked was the percentage of roles shared among the users:

Users share 15% of the roles: 8 roles shared

Users share 30% of the roles: 12 roles shared

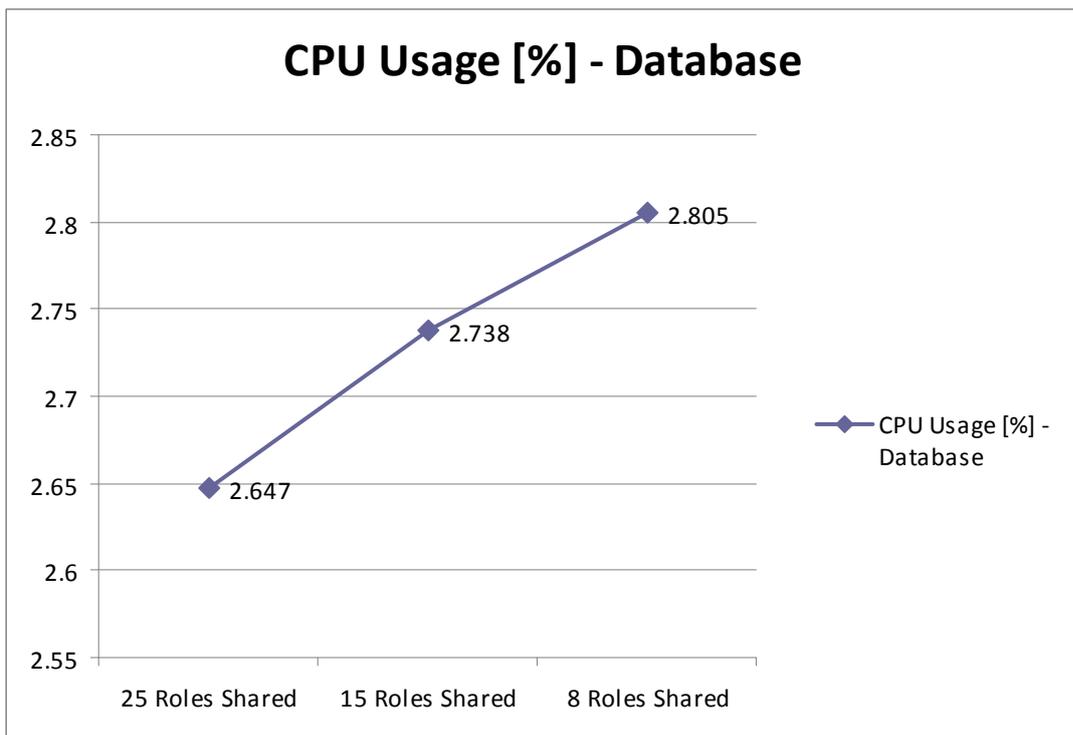
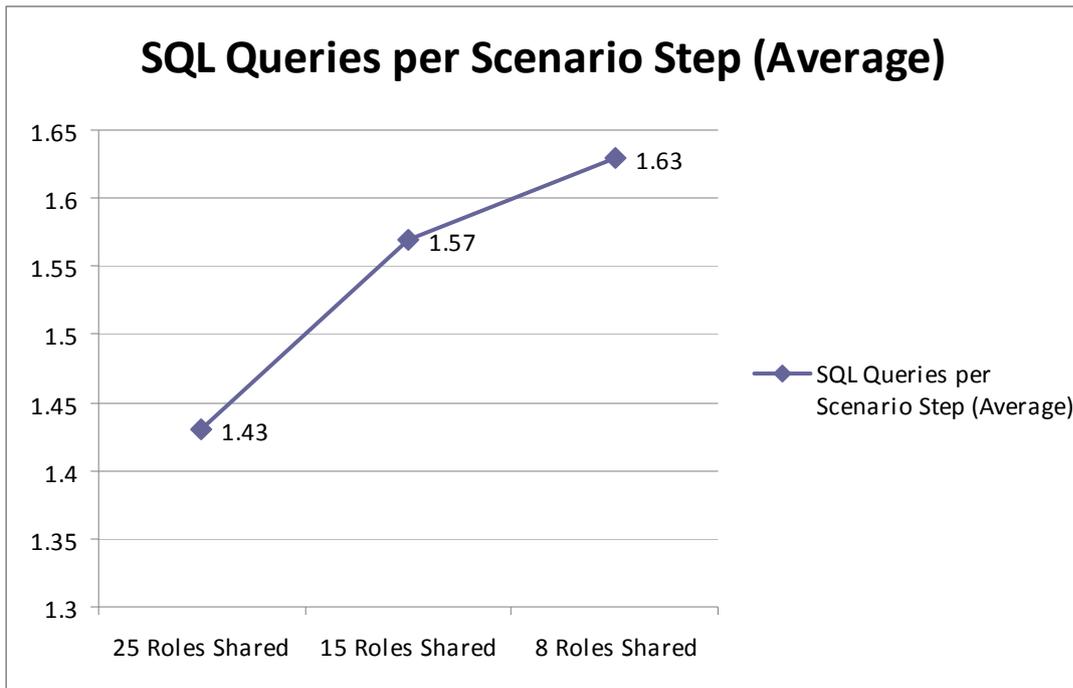
Users share 50% of the roles: 25 roles shared

The average response times, memory per transaction, and CPU of the server stayed within the same range in all three tests.

Conclusion

When more roles were shared, there were fewer trips to the database, thus reducing the database CPU consumption.

In all cases with 50 roles per user, the different role-sharing percentages had no effect on the response time, for this specific benchmark.

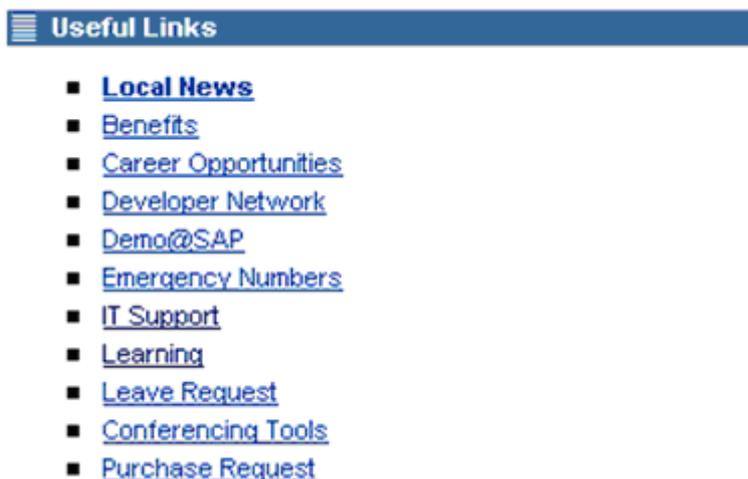


5. Page Content

The more content a portal page contains, the longer it takes to load the page. Furthermore, the isolation method of the iViews contained in a page can influence performance. (More information: [Isolation Method of iViews](#) in the SAP Help Portal).

Rather than overloading a page with content, it is advisable to distribute information among a number of pages. This enhances performance and avoids the need of scrolling down a long page screen. Keep the welcome page simple; use it to provide useful information to user (for example, news). Typically, organizations do not require heavy applications on the start page, rather links to most frequently used functionality. Make applications available using standard navigation.

A typical start page might look like the following illustration:



- [Local News](#)
- [Benefits](#)
- [Career Opportunities](#)
- [Developer Network](#)
- [Demo@SAP](#)
- [Emergency Numbers](#)
- [IT Support](#)
- [Learning](#)
- [Leave Request](#)
- [Conferencing Tools](#)
- [Purchase Request](#)

6. Hardware Sizing

In this chapter, general concepts of performance, scalability and sizing are discussed. A detailed description of the sizing for SAP NetWeaver Portal can be found in the sizing paper on the SMP (<http://service.sap.com/sizing> → *Sizing Guidelines* → *Solutions & Platforms* → *Sizing SAP NetWeaver Portal*)

6.1 Importance of Hardware Sizing to Performance

The performance of an application depends not only on the software configuration, but also on the hardware configuration. The purpose of hardware sizing is to determine the most suitable hardware configuration to avoid possible performance bottlenecks caused by a shortage of hardware resources.

 Note

Proper hardware configuration ensures reasonable system response times under given system load, but no specific system response times can be derived.

6.2 Sizing and Scalability

The SAP sizing process provides a way of predicting the hardware resource requirements of software applications used in an organization, which, in turn, depend on the requirements of the company business processes. The means for determining the hardware requirements consist of two major parts:

- Creation of sizing guidelines

Sizing guidelines are based on the results of performance tests and analysis of software applications, called the sizing measurements.
- Determining how to use these sizing guidelines in customer sizing projects

6.2.1 Understanding Scalability

Scalability is a prerequisite for proper sizing and it must be proven before reliable sizing can be determined. Scalability refers to the predictable resource consumption of a software application under different system loads while the response time remains within a reasonable range.

Scalability is related to many factors, such as data volume and hardware adaptability. The most prominent scalability dimension is the number of concurrent users actively working in a system at the same time. For this reason, it is essential to conduct scalability tests during sizing measurements to ensure requisite scalability for sizing. The following example illustrates how to prove the CPU scalability regarding number of concurrent users.

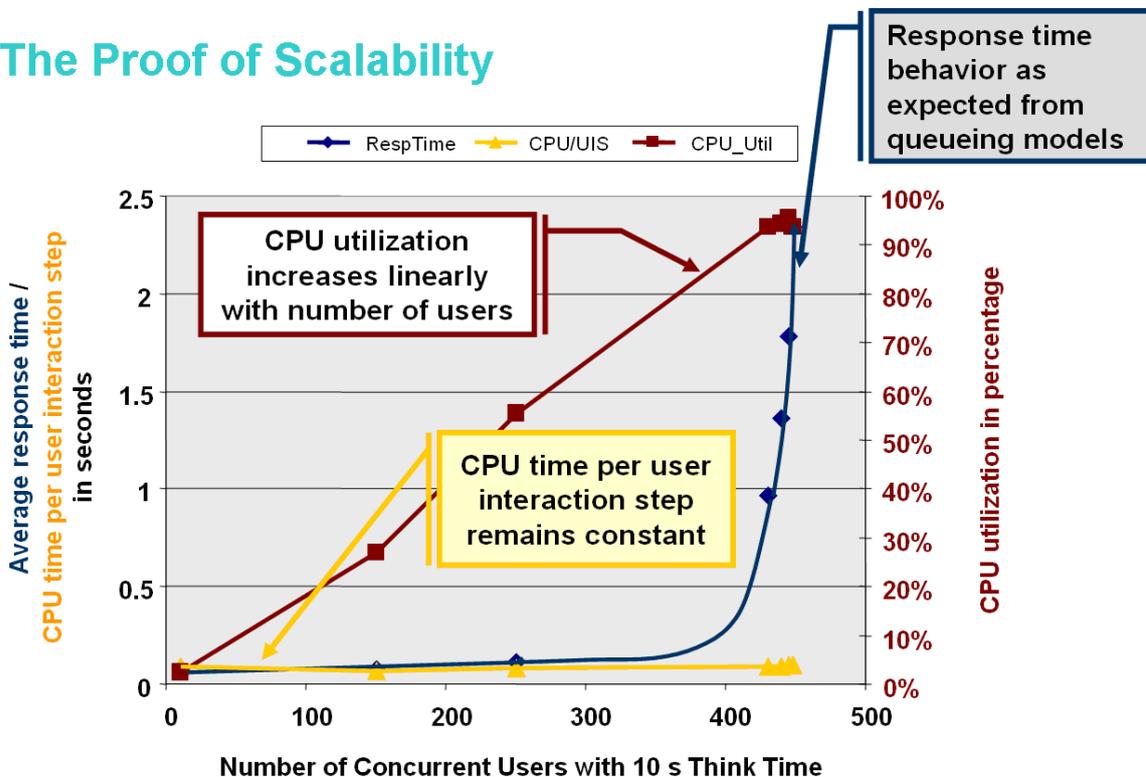
Proving Scalability

After carefully analyzing critical business processes, a scenario is determined encompassing a number of user interaction steps. The next step is the simulation of an increasing number of concurrent users executing this scenario at a constant think time and measure the following KPIs:

- CPU time per user interaction step
- CPU utilization
- Response time

Putting the results of these measurements together in a chart enables examination of CPU scalability. The following chart shows the results of Multi-user load tests using an SAP NetWeaver Portal Benchmark EP-ESS (Employee Self-Service) to prove CPU scalability.

The Proof of Scalability



- CPU time per user interaction step remains constant over increasing numbers of concurrent users.

This corresponds to the fundamental assumption in the queuing theory models that the service time of a specific request is independent of current system utilization. Otherwise, if the CPU time per user interaction step increases, this would cause an over-linear increase of CPU utilization and an exponential increase of response time (called a “busy waiting” issue) with regard to CPU scalability.

- CPU utilization increases linearly with the number of concurrent user.

This behavior demonstrates that the software is able to utilize available CPU resources in an efficient way. If the CPU utilization does not increase linearly with the number of concurrent users, the response time increases rapidly; this is a CPU scalability serialization issue.

- Response time behaviors as expected by the queuing theory model.

As seen in the graph, and according to the M/M/n model, the response time should at first remain fairly constant as the number of users increase, then linearly increase, and, finally, end in an exponential increase, where the CPU utilization is close to 100%.

For the example shown in figure 1, the relationship of CPU usage to the number of users is predictable for a considerable number of users. Scalability is proven and sizing guidelines can be applied. Sizing guidelines cannot be applied if the number of users to CPU relationship is unpredictable.

6.3 Create Sizing Guidelines

Sizing guidelines represent the hardware resource consumption behavior of a software application as it relates to business process parameters, which are called influencing factors. To identify influencing factors it is necessary to understand the business process, as well as the software architecture and its implementation.

Creating the guidelines is an iterative process of experimental measurement and analysis. The result of this process is that a set of influencing factors and a related set of test cases are defined. For the defined test cases, it is necessary to measure the following sizing KPIs to compile the sizing formulas:

- CPU Consumption
- Memory
- Front-End Network Requirements

6.3.1 CPU Consumption

6.3.1.1 CPU Sizing Formula in Terms of CPU Time

Typically CPU time is measured per user interaction step in relation to the influencing factors. For portal sizing, we measure the CPU time of a portal navigation step. For example, the influencing factors of a portal navigation step are the number and types of the iViews displayed on a target portal page of a portal navigation step. If an average CPU time per user interaction step has been determined, then the CPU sizing formula is as follows:

$$\text{numberOfCPUNeeded} = \text{CPUTimePerUserInteractionStep} * \text{numberOfConcurrentUsers} / ((\text{thinkTime} + \text{responseTime}) * \text{CPUUtilization})$$

6.3.1.2 CPU Sizing Formula in Terms of SAPS

In CPU sizing, we use a hardware platform-independent measuring unit called SAPS (SAP Application Performance Standard) for measuring CPU processing power related to business processes. The processing power needed to complete 2,000 fully processed sales order items in one hour is defined as 100 SAPS. This business process is defined by the SAP Sales and Distribution (SD) Standard Application Benchmark. By using the SD benchmark, the SAPS number of any CPU type can be determined. The SAPS numbers of different hardware configurations can be found in certified SAP standard application benchmarks at <http://www.sap.com/benchmark>. In terms of SAPS, the CPU sizing formula can be rewritten as follows:

$$\text{totalSAPSNeeded} = \text{SAPSPerCPUUsedInTest} * \text{CPUTimePerUserInteractionStep} * \text{numberOfConcurrentUsers} / ((\text{thinkTime} + \text{responseTime}) * \text{CPUUtilization})$$

6.3.2 Memory Consumption

Since SAP NetWeaver Portal is a Java application, three Java memory KPIs (introduced in [Cheng, X., TechEd06] and [Cheng, X., TechEd 07]) are to be considered:

- The framework space

Amount of live objects in the Java heap in a warm system, when no user is logged on. The framework space depends on the deployed, stated, and used applications, and their cache content that is shared by all users.

- The user session space
Amount of live objects making up the user context, which cannot be shared by other users
- The processing space
The average garbage-collected memory per user interaction step, representing the dynamic Java memory consumption.

 Note

It is worth keeping in mind that memory consumption is an important factor with Java applications because of garbage collection.

More information: [Impact of Java Garbage Collection \(GC\) on Performance](#) on page 4

If the user session space, which is the static part, is dominating in memory consumption, then the memory sizing formula is:

$$totalRequiredMemory = userSessionSpace * numberOfConcurrentUsers$$

If the processing space, which is dynamic, is dominating, a possible memory sizing model could be determined by the CPU.

 Recommendation

In sizing SAP NetWeaver Portal, we recommend 1 GB physical memory for every 300 SAPS CPU power.

6.3.3 Front-End Network Requirements

The two main-front end KPIs are:

- Number of network round trips
- Data volume transferred per user interaction step

For bandwidth sizing, the following formula can be applied:

$$totalBandwidth = safetyFactor * transferredDataVolumePerUserInteractionStep * numberOfConcurrentUsers / (thinkTime + responseTime)$$

The *safetyFactor* can be determined by taking the following factors into account:

- Protocol overhead
- Feasible utilization of a network connection (More information: <http://service.sap.com/sizing> → *Sizing Guidelines* → *Solutions & Platform* → *Front-end Network Requirements for SAP Solutions*)

6.4 Apply Sizing Guidelines

The implementation of a customer sizing project takes place within the scope of implementing a software application at a customer site. Using the application-specific sizing guidelines, technical consultants start by working with the business department, the implementation project team, and the hardware vendor to determine the following:

- The application usage profile – the concrete values of the influencing factors
- The user activity profile – the number of concurrent users and average think time, derived from the business process to be implemented.

These factors comprise the sizing input values necessary for calculating hardware resources. With the values for these factors, the formulas of the sizing guidelines can be applied to calculate the required hardware resources. Taking into consideration the eventual need for further system landscaping

requirements, like high availability, for example, different hardware vendors can provide their proposals for the target hardware configurations.

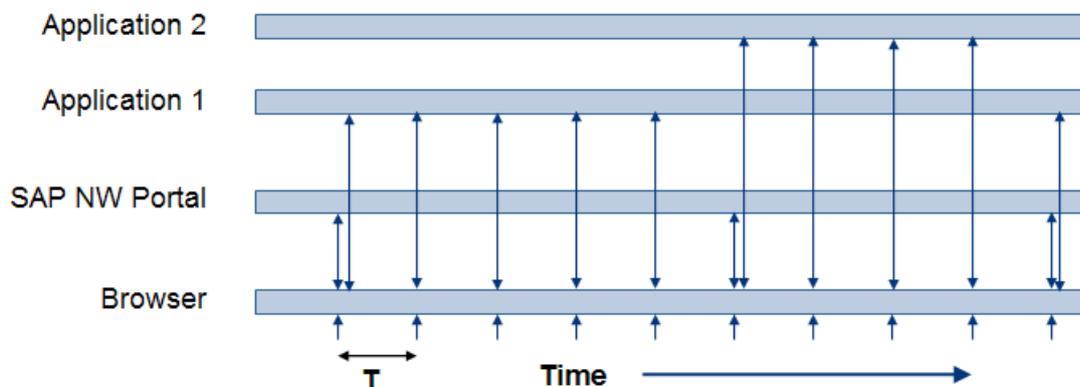
Example

Consider the CPU sizing of SAP NetWeaver Portal. To determine the application usage profile, we consider the typical portal page, in terms of number and types of iViews displayed on that page. With this information, a corresponding value of the term *SAPSPerCPUUsedInTest * CPUTimePerUserInteractionStep* can be found in the sizing guideline, and then the [CPU sizing formula](#) (on page 47) can be applied, where we assume an average CPU utilization of 65% for production systems.

SAP NetWeaver Portal, as part of the presentation layer for SAP applications, supports server-applications. Therefore, when sizing a portal integrated application scenario, often only an overall usage profile and user activity profile can be derived from the business process level. In this case, the overall load profile needs to be converted into application-specific load profiles in order to be able to apply application-specific sizing formulas. The conversion depends on the interaction patterns between the applications.

In figure 2, an entity interaction diagram is shown for a portal scenario integrating two applications. Two different types of user interaction are shown in this scenario: portal top-level navigation to launch applications, and user interactions within an application. As shown in figure 2, the user interaction steps within an application bypass the portal, and so they don't generate any load on the portal.

Given that the average think time of a user is T , and the user needs to execute an average five user interaction steps in application 1 and four user interaction steps in application 2, the average think time in the portal for applications 1 and 2 is $9*T/2$, $9*T/5$, and $9*T/4$, respectively. The following illustration shows different think times in a portal integrated application scenario.



6.5 Initial Sizing and Expert Sizing

Initial Sizing

Initial sizing recommendations provide sizing statements related to SAP standard delivery of software applications. Since the sizing results are hardware platform-independent and can be derived directly from business requirements, initial sizing can be used in presales cases to help customers to do a rough hardware budget planning.

Initial sizing of SAP NetWeaver Portal is supported by the Quick Sizer tool, available for SAP customers at <http://service.sap.com/sizing> → *Quick Sizer Tool*. In many portal-integrated application scenarios, especially when the Web Dynpro Java-based presentation layer is running on the same SAP NetWeaver Web Application Server Java as the Portal, the portal sizing is already included in the sizing of the application scenario. Using the QuickSizer, you can also size the portal

component specifically, when, for example, the portal needs to run on a separate server to integrate third party applications.

Expert Sizing

When software applications are highly modified, sizing measurements need to be conducted at the customer site to adjust the sizing guidelines. This sizing approach is referred to as expert sizing. Essentially, this is the same procedure described in the section [Create Sizing Guidelines](#) on page 47.

6.6 Sizing Verification

For mission critical software projects, custom load tests can be performed to verify the sizing results, before going live. By simulating the specific system load, it can be demonstrated that the installed system landscape is able to fulfill the specified performance requirements.

During the productive operation of software applications, sizing verification can be done by comparing the sizing assumptions and results with the real system load and behavior. By analyzing system monitoring data, the real application usage profile and the user activity profile can be determined and compared to the sizing assumptions. Other aspects sizing can also be considered, such as sized SAPS compared to installed SAPS, and further compared to SAPS that are actually used. Sizing verification is not only important to adjust configuration of the system landscape, but it is also helpful to better understand user behavior, which would provide better sizing assumptions for future sizing projects.

6.7 References to Sizing Documentation

[Janssen, S; Marquard, U.]
Sizing SAP Systems (Galileo Press GmbH, 2007)

[Cheng, X., TechEd 06]
<http://service.sap.com/performance> -> Media Library -> "How-to: Best Practices for Java Performance & Load Tests"

[Cheng, X., TechEd 07]
<http://service.sap.com/performance> -> Media Library -> "How-to: Best Practices for Java Performance Analysis"

[Front end network sizing paper]
<http://service.sap.com/sizing> → *Sizing Guidelines* → *Solutions & Platform* → *Front-end Network Requirements for SAP Solutions*