

## SAP Enterprise Portal 6.0 Performance Test



### **Applies To:**

SAP Enterprise Portal 6.0 SP14, SAP NetWeaver 2004

### **Summary**

This article is based on our activity of testing and tuning the Performance of SAP Enterprise Portal 6.0, SP14 in the Minimal landscape configuration on Solaris 9 running on Sun Fire v1280 server. This article provides few simple steps to convert the out of the box setup to a well tuned system.

**By:** Vasanth Bhat

**Company:** Sun Microsystems India Ltd.

**Date:** 15 February 2006.

## Table of Contents

Executive Summary .....	3
Chapter 1 : Introduction .....	4
Chapter 2 : General & Technical Descriptions .....	5
SAP Enterprise Portal:.....	5
SAP J2EE Web Application Server: .....	6
Chapter 3 : Scenario Details .....	7
Chapter 4 : Test Environment .....	9
4.1 Hardware setup.....	9
4.2 Software Setup .....	11
4.3 Performance Test Procedure .....	11
4.4 Performance Tuning Steps.....	13
Chapter 5 : Results.....	17
Chapter 6 : Conclusions .....	19
Chapter 7: References.....	20

## Executive Summary

SAP Enterprise Portal implementations typically begin with an out of the box standard installation. The out of the box installations typically perform poorly. As a final part of the implementation, the system goes through a lot of testing and Tuning at various levels like the application, J2EE Engine, JVM, Database and OS. This is required for higher levels of performance and scalability. This is often a very complex and expensive process. The expertise and technical information for this have to be obtained from a lot of sources. This can add significantly to the Total Cost of Ownership (TCO)

This performance exercise was conducted with an aim to help the customer in reduce the TCO for setting up a well tuned implementation. Keeping the TCO in mind, the Minimal Landscape was chosen, where in all the NetWeaver components are on the same server and share the same Web Application Server and Database Instance. This exercise also aims to provide consolidated set of information to transform the out of the box Enterprise Portal setup to a well tuned system.

In SAP Enterprise Portal, certain iView settings like cache settings and isolation method have an overall effect on performance and stability. This performance exercise also attempts to study their impact on the iView performance.

The overall result of the performance tests was that the performance of out of the box installation of SAP Enterprise Portal in a NetWeaver Stack can be improved considerably with proper selection of iView cache settings and a few simple tuning steps.

# Chapter 1

## Introduction

This document shows the tuned implementation of SAP Enterprise Portal 6.0 SP10, SAP J2EE (Add In) Web Application server 6.40 on Oracle 9.2.0.5.

Post implementation, there are some tuning steps needed for the system to perform at optimum levels. By optimum levels, we mean:

- (a) Scale the number of users supported by the system.
- (b) Improve the response times for the users of the system at the same time.
- (c) Better utilization of the system and hardware resources at hand.

The goals of this exercise are:

- Reduce the total cost of ownership for SAP NetWeaver by decreasing the time required to transform an out of the box installation to a well tuned system. Typically an out of the box installation needs a lot of performance tuning and sizing at various levels (OS, SAP software, J2EE, database). This tuning information is spread over various data sources like SAP Notes, SAP Service Marketplace, Sun blueprints and Oracle documentation. The time required to peruse this information and implement all steps to achieve an optimal configuration can take many weeks. The primary goal of this exercise is to consolidate this information in one document.
- Validation of practical customer scenario with all NetWeaver components installed in a single stack
- To validate the performance of the NetWeaver single stack on a Production Ready Server.
- Study the performance of iView execution with different cache level and isolation method settings. The settings for these properties have an overall effect on portal performance and stability.

## Chapter 2

### General & Technical Descriptions

#### SAP Enterprise Portal:

SAP® Enterprise Portal is a key component of the SAP NetWeaver™ platform. SAP Enterprise Portal is a comprehensive portal solution, providing a complete portal infrastructure along with bundled knowledge management and collaboration capabilities. It provides people-centric integration of all types of enterprise information, including SAP and third-party applications, structured and unstructured data, and Web content. It is based on open standards such as Web services and Java 2 Platform, Enterprise Edition (J2EE).

Business content delivered with SAP Enterprise Portal speeds portal implementation and reduces the cost of integrating existing IT systems. SAP Enterprise Portal provides employees, supply chain partners, customers, and other user communities with immediate, secure, and role-based access to key information and applications across the extended enterprise. As part of SAP Enterprise Portal, the portal infrastructure provides secure access to structured data coming from any kind of business application, such as data warehouses, databases, and SAP or non-SAP applications. Because information and applications are unified through the portal, users can identify and address business issues faster, more effectively, and at lower cost – creating measurable benefits and strategic advantages.

SAP Enterprise Portal components and services provide content as iViews. iViews are programs that run in the portal client, such as a Web browser. As the key technology of the portal infrastructure, iViews allow access for reading purposes or to carry out an operation, such as submitting a form for supplies. A set of portal services is provided that enables iViews to interact with each other. Examples include portal navigation (e.g. dynamic navigation or the patented Drag&Relate technology), portal eventing, and WorkProtect mode.

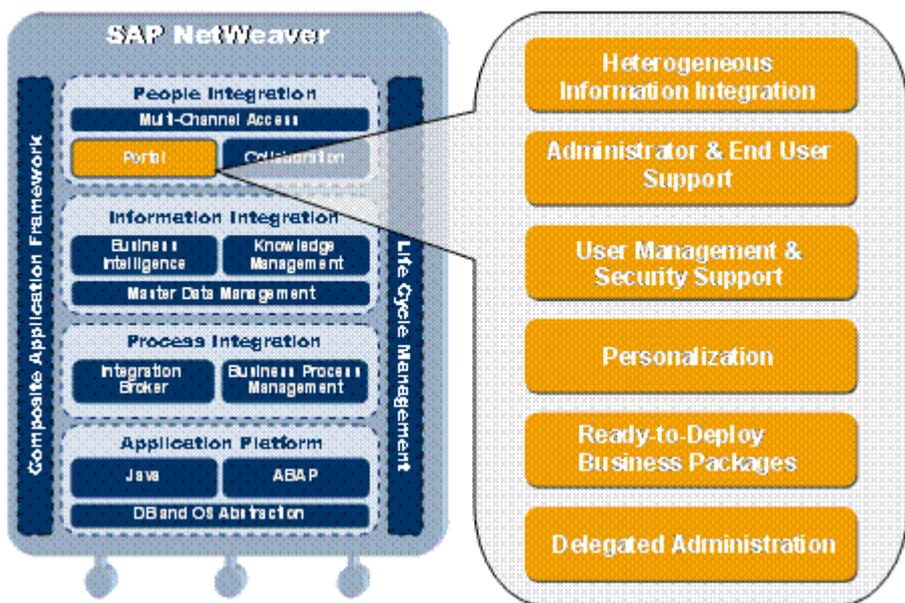


Figure 2.1 - SAP Enterprise Portal Components

## SAP J2EE Web Application Server:

- Integrates the ABAP and Java personalities in one application server – a homogeneous infrastructure for J2EE-based and ABAP-based applications – supporting the existing ABAP applications - ABAP-based, Java-based and Web-based application development – OS and DB portability for both ABAP-based and Java-based applications - high performing internal ABAP-Java communication.
- Brings together the benefits of a proven, scalable and reliable infrastructure with the interoperability and flexibility of Web Services technology – SAP Web AS may act as both the Web Services client and - server for easily integrating existing and new enterprise applications behind a firewall as well as driving collaborative scenarios that cross company boundaries.
- Is the powerful Java platform J2EE certified application server.
- SAP enhancements enable the enterprise quality application development in Java

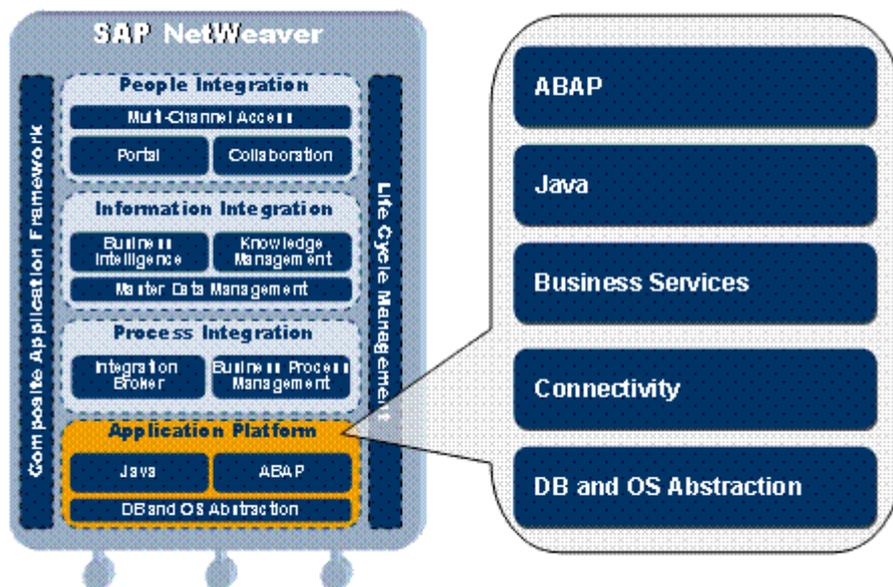


Figure 2.2 – SAP Web Application Server components

## Chapter 3

### Scenario Details

Typically an enterprise has information integrated from various data sources. This information is channeled through iViews in the portal. Some of this information can be dynamic (i.e. the information changes frequently or it comes at real time from a transaction based database) and some information can be relatively static (i.e. the information does not change so frequently). Examples of static information can be newsletters published weekly or monthly, policy documents, online help for various business practices etc.

To improve the performance of the static content being channeled into the portal via iViews, we can improve the user experience by decreasing the response times of these iViews. This can be done via caching. Caching mechanism stores the content retrieved by an iView request in memory for a preconfigured amount of time. Successive identical requests first access the cache, rather than resubmitting the query to the data source, and if the cache has not expired, the iView displays the information contained there. Caching improves response time and overall system performance by reducing the load on the information source.

While we usually use the portal cache to improve the response time of content that does not change frequently, applying caching to content that is updated often is also worthwhile, because it still takes time to generate the content.

The browser has a similar mechanism to that of the portal cache, whereby it can store content on the client side and display it as long as it is valid. This further improves response time by reducing the load on the portal, because the browser uses its stored content instead of requesting it from the portal.

In our load testing scenario, we have the use case where we wanted to test the efficiency of the different types of caching levels for the iViews. We tested all the 4 types of caching mechanisms one after the other, in one scenario itself.

Property Name	Property ID	Admin	End User	Object Type	Description
Cache Validity Period (msecs)	ValidityPeriod	Editable	Hidden	iView, page	Validity period (in milliseconds) of content in cache before it is refreshed
Cache Level	CachingLevel	Editable	Hidden	iView, page	Defines level of caching: shared by everyone, or cache per user or browser session Values: Shared, User, Session (explained in detail below)

The caching mechanism is part of the portal runtime and operates according to specific application and iView settings. Activating the various levels of caching entails configuring iView properties that:

- Allow caching in the client browser
- Set the cache level
- Set the validity period of the cache

- Automatically refresh iViews from cache

### **Allow Caching in the client browser**

This property is used to define whether an iView's content can be cached in the client browser, as well as in the portal cache. If caching an iView on the portal improves performance, then allowing client-side caching further enhances it.

The only drawback to client-side caching concerns iViews that display sensitive or private content, which you would not want to store locally on the browser machine.

The property values are:

#### **True (default)**

Enable browser cache settings to be defined per iView

#### **False**

Disable the browser cache for all iViews, regardless of individual iView settings

### **Set the Cache level**

You determine the cache level according to the content of the iView, the speed of retrieval, and the configured time of loading. The four cache levels are:

#### **None (default)**

iView content is never cached. Each time the iView is launched, it accesses the content source. This level is appropriate for iViews retrieving information that is constantly changing and being updated, such as a news bulletin iView, or a ticker to a stock exchange.

#### **Session**

iView content is cached for the current session. Each new logon empties the cache and the iView must access the source. An iView retrieving secured content would likely use this level.

#### **User**

iView content is cached per user. New users who launch the iView receive content from the data source. An iView representing a user's daily tasks is appropriate for this cache level.

#### **Shared**

iView content retrieved by first user is cached for all users. This level is suitable for iViews displaying information relevant to all users, such as company announcements.

### **Set the validity period of the cache.**

This setting defines the expiration time of iView content stored in cache. Once the cache has expired, the iView must access the data source for its content.

### **Isolation Method of iViews**

The Isolation Method property of iViews defines how an iView or page is encoded in a page. It determines whether iView content is collected at the server or at the client, and how the PageBuilder component displays the content. The value of this property can affect the performance of a page.

iViews can have one of the following isolation methods:

### **Embedded Isolation Method**

- iView content is collected at the server side and sent to the client with the entire page.
- One request from the client to the server causes the server to reload all the iViews on the page.
- PageBuilder has no control over the appearance of the iView content.

Content is presented “as is” in the browser  
iView size is dependent on the content  
Height not controlled by the page  
Page does not add scrollbars to iViews

- Supports both server-side and client-side events

### **URL Isolation Method**

- iView is written into an iFrame, which is inserted into a container on the page. Each iFrame works independently.
- iViews are not encoded in the page HTML

PageBuilder generates an iFrame and sets the source attribute to point to the iView URL  
iView’s content is generated after an additional server request  
PageBuilder writes content into the iFrame on the client side

- Interactions and reload actions on the iView do not affect the rest of the page
- PageBuilder has full control over the appearance of the iView content.

Page determines iFrame size  
Page can add vertical and horizontal scrollbars to iFrame

- Supports only client-side events

## Chapter 4

### Test Environment

#### 4.1 Hardware setup

The entire test was run on a single SunFire v1280 Server. The server has the following configuration.

Model Sun Fire v1280

CPU's 12 UltraSPARC™ IIIcu 64 bit@900-MHz, 8 MB L2 cache

Memory 24 GB

Storage Sun StorEdge 3510 FC Array (12 \*73GB).

The Sun Fire V1280 server gives enterprises the compute capabilities of large systems in a dense, 12U rack-mountable form factor. The SunFire v1280 server also supports a number of features to improve the availability of systems running mission-critical or datacenter applications, including redundant, hot-pluggable components, optional redundant paths to external storage, and support for clustering. The SunFire v1280 server is an ideal platform for creating large network infrastructure, transaction processing systems, and compute farms.



**Figure 4.1** : SunFire v1280 server.

## 4.2 Software Setup

The testing environment was setup with the following software components.

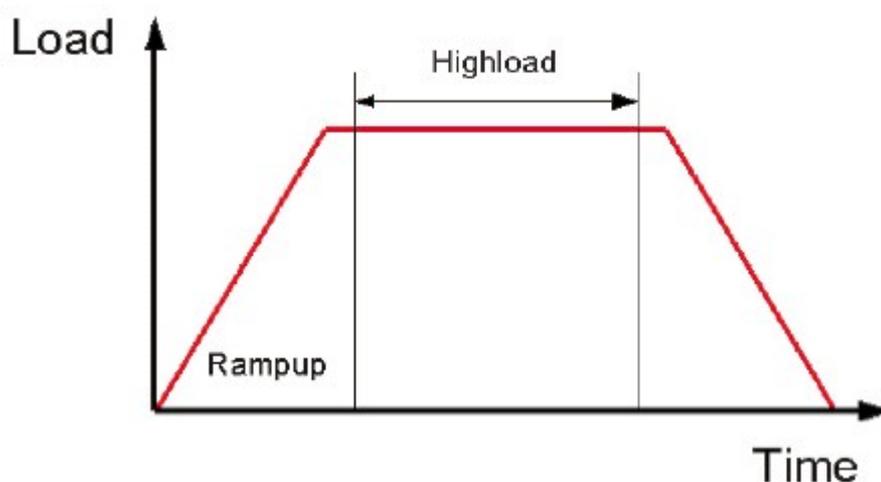
OS	Solaris 9
Database	Oracle 9.2.05
SAP Kernel	Web Application Server 6.40, Patch 57
SAP J2EE	SAP J2EE 6.40, SP14
NetWeaver Components	Enterprise Portal 6.0 SP14 Knowledge Management, Collaboration eXchange Infrastructure, 3.0 SP14 Business Warehousing/Business Intelligence 3.5 TREX SP14 Adobe Document Engine

The entire NetWeaver stack was setup on a single instance of J2EE Add-In installation of Web Application Server 6.40. The NetWeaver stack was patched to service pack 10.

## 4.3 Performance Test Procedure

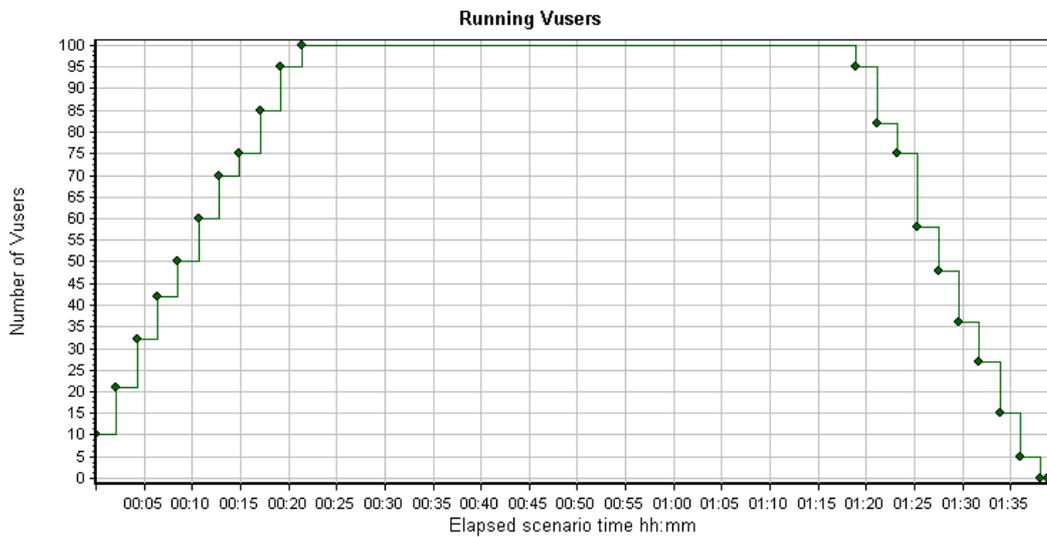
The SAP Enterprise Portal performance tests were carried out using the Mercury LoadRunner™ tool.

The load profile used in the tests followed the procedure used in the well known SAP benchmarks. The test started with the initial load and then gradually increased to full load during the ramp-up period. The execution continued at full load for a fixed amount of time and then the load was ramped down gradually.



**Figure 4.2 : Load profile of the Performance test.**

The following graph displays the number of users that executed the scenario, and their status, during each second of the performance test. This graph is useful for determining the user load on the server at any given moment.



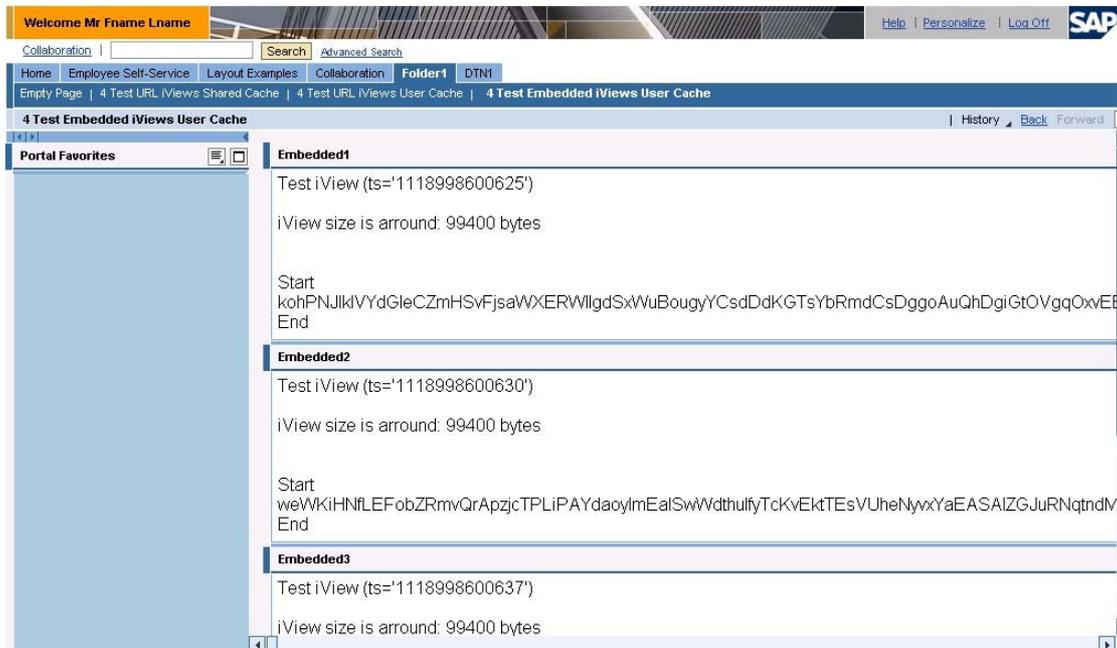
**Figure 4.3 : Load profile of the 100 user Performance test**

The following steps were executed by each of the LoadRunner generated Vusers (Virtual User).

1. Logon to the SAP Enterprise Portal.
2. Select link on top navigation panel.
3. Click on the link to load Empty page in an iView.
4. Click on the link to load page containing four iViews using Shared Cache.
5. Four Shared Cache iViews are executed with each generating 100KB of content.
6. Click on the link to load page containing four iViews using User Cache.
7. Four User Cache iViews are executed with each generating 100KB of content
8. Click on the link to load page containing four iViews using Embedded Cache.
9. Four Embedded Cache iViews are executed with each generating 100KB of content.
10. Perform above steps 2-9 iteratively till ramp down.

## 11. Logoff from Portal

The following is a screen shot of the layout used for the Performance test.



The following Key Performance Indicators (KPI) are monitored during the execution of the tests.

- **LoadRunner End-to-End user response time [sec]** : It measured from the end-user perspective. A good example of response time is a measurement of the time between when a user clicks the Login button and when the subsequent page is fully loaded. Summation of user response time for all transactions (Login --> Action Transactions --> Logoff) is End-to-End user response time.
- **System load [number of users with stable response times]** : Number of simultaneous Users with stable and acceptable response time for all transactions.
- **Stability (MTBF) [hours]** : Number of hours for which the Web site or Application can handle an increased number of users (when CPU usage is > 60 %) before failure. (Mean Time Before Failure)
- **Scalability [number of nodes]** : It is the ability of a Web site or application to handle an increased number of users while still meeting performance goals. A Web site or application is often said to scale gracefully if its performance degrades slowly when predicted user levels exceed expectations as opposed to becoming unusable or crashing.
- **Average CPU usage in %** : Average CPU usage during the Performance test run.

## 4.4 Performance Tuning Steps

The performance tuning was done at the SAP WAS J2EE Engine, Database and the OS Level.

### 4.4.1 SAP WAS J2EE Engine

- Server Instance Setup

The number of server process in the J2EE Engine is an important factor for the scalability and overall performance. The J2EE Engine dispatcher balances the load among the server process during the execution of the performance test.

We found that using multiple J2EE server instances was a suitable approach to make the best use of the large number of CPUs. The J2EE engine was configured with 4 server instances. These J2EE server instances were configured as standalone cluster. The J2EE servers process the most of the workload in this scenario. Having many processes handling smaller chunks of the work as opposed to have fewer processes handling larger chunks each was found to be the best approach to achieve scalability.

- Server Instance Configuration

SAP J2EE Engine server instances also can be configured with different processing threads. The thread parameters can be configured globally for all the server instances through the Application Thread Manager in the J2EE Engine configuration tool. On the same lines the Database connection pool parameters can be configured through the Configuration Manager in the J2EE Engine configuration tool. The table below gives the default value and the modified values for the tuned parameters.

	Parameter Name	Default Value	Tuned Value
1.	Initial Thread Count	40	50
2.	Max Thread Count	40	200
3.	MaxRQSize	5000	10000
4.	MaxFOQSize	5000	10000
5.	RDBMS Initial Connections	1	5
6.	RDBMS Max Connections	5	25

- Java VM Tuning for J2EE Engine

The Java VM Parameters for the J2EE Engine can be tuned using J2EE Engine Visual Administrator. The table below gives the default values and the modified values for the tuned parameters.

	Parameter	Default Value	Tuned Value
	Server.MaxHeapSize	1024	2048
	Java Max Heap Size	-Xmx512M	-Xmx2048M
	Java Min Heap Size	-Xms512M	-Xms2048M
	NewSize	170M	512M
	MaxNewSize	170M	512M

The GC logs were collected and analyzed for 12 hour runs. Our analysis showed 99.99% throughput and negligible total GC load. The following is the final GC settings.

```
-verbose:gc
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
-XX:+UseParNewGC
-Xms2048m
-Xmx2048m
-XX:MaxNewSize=512M
-XX:NewSize=512M
-XX:+DisableExplicitGC
-XX:MaxPermSize=192M
-XX:PermSize=192M
-XX:SurvivorRatio=2
-XX:TargetSurvivorRatio=90
-XX:+UseLWPSynchronization
```

#### 4.4.2 Operating System Tuning

- Kernel Parameters

The system was tuned with the following kernel parameters in the /etc/system

```
set msgsys:msginfo_msgmax=8192
set msgsys:msginfo_msgmnb=16384
set msgsys:msginfo_msgmni=50
set msgsys:msginfo_msgtql=2048
set semsys:seminfo_semmns=16384
set semsys:seminfo_semmni=1024
set semsys:seminfo_semmnu=4096
set semsys:seminfo_semopm=100
```

```
set semsys:seminfo_semume=64
set semsys:seminfo_semmsl=2048
set shmsys:shminfo_shmmax=0xfffffffffffff
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=10
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=10
set rlim_fd_max=8192
```

- Network Tuning

Following tuning was carried out at the network level using the Solaris “nnd” command.

```
nnd -set /dev/tcp tcp_conn_hash_size 32768
nnd -set /dev/tcp tcp_time_wait_interval 60000
nnd -set /dev/tcp tcp_conn_req_max_q 1024
nnd -set /dev/tcp tcp_conn_req_max_q0 4096
nnd -set /dev/tcp tcp_smallest_anon_port 1024
```

## Chapter 5

### Results

The results are summarized in the following graphs.

The following graph shows the response times in seconds for the transactions using different caching mechanisms for multiple test runs.

Transaction Name	100 users <b>Out of Box</b>	100 users <b>Tuned</b>	500 users <b>Well Tuned</b>	1000 users <b>Well Tuned</b>
%CPU Utilization	28	34	48	53
lr_logon	49.42	3.321	0.262	0.301
lr_MainMenu	13.78	0.544	0.043	0.108
lr_emptypage	14.28	1.184	0.091	2.146
lr_UserCache	21.26	1.551	0.097	2.127
lr_SharedCache	21.00	1.430	0.392	3.366
lr_EmbeddedUserC ache	32.61	2.119	0.093	2.265
lr_Logoff	11.75	1.061	0.169	1.470
End to End Response (Total)	164.10	11.210	1.147	11.78

**Figure 5.1 - iView Execution Response times**

lr\_logon: This transaction represents the action for the user logging into the SAP Enterprise Portal.

lr\_SharedCache: This transaction represents the execution of iViews which use “Shared Cache” as the Caching Level and “URL” Isolation Method.

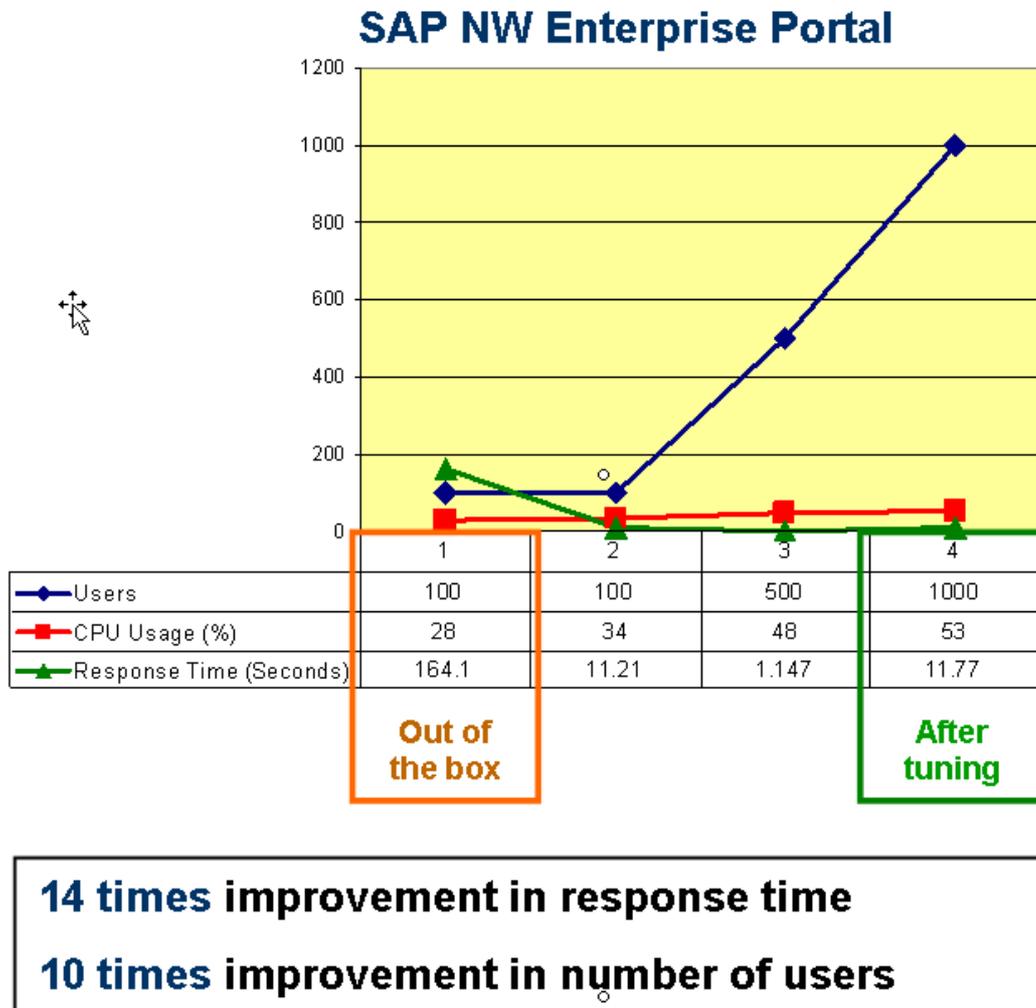
lr\_UserCache: This transaction represents the execution of iViews which use “User Cache” as the caching Level and “URL” Isolation Method.

lr\_UserEmbeddedCache: This transaction represents the execution of iViews which use “User Cache” as the caching Level and “Embedded” Isolation Method.

lr\_logoff: This transaction represents the action for the user logging out of the SAP Enterprise Portal

The following graph shows the Key Performance indicators during each of the

performance test runs and highlights the effect of the Tuning on the overall Performance. The response times are the end to end user transaction times, which is the total of the response times for the above transactions involved in each user end to end action.



**Figure 5.2 – Effect of Tuning on the overall Performance**

**Important Note :** Due to resource constraints we couldn't load the server to its fullest capacity. At its maximum the server was loaded with 1000 users concurrently executing transactions resulting in average CPU Utilization of 53%

## Chapter 6

### Conclusions

The results of the above performance exercise shows that performance of out of the box installation of SAP Enterprise Portal in a NetWeaver Stack can be improved considerably with a few simple tuning steps.

In our tests we could improve the overall response times by about 14 times with above simple tuning steps. The tuning steps also increased the scalability of the system. We were able to support about 10 times more number of users at approximately about 50% CPU utilization.

The result also showed that the certain iView settings like cache settings and isolation method have an overall effect on performance and stability. The response times varied for the execution of same set of iViews. This is shows that the choice of the settings is important. For example, use of share cache will give better performance for static content, but may not be suitable for dynamic content.

## Chapter 7

### References

1. SAP NetWeaver '04 How-To Guide : Tune Performance of Portal Platform
2. SAP note 785927 about SAP software on UNIX: OS Dependencies
3. SAP note 723909, JVM settings for J2EE 6.30 and 6.40
4. SAP Enterprise Portal documentation, [http://help.sap.com/saphelp\\_nw04/](http://help.sap.com/saphelp_nw04/)
5. Sun blueprints <http://www.sun.com/blueprints/>
6. Sun documentation <http://docs.sun.com>

### Author Bio

The author is a Lead Engineer in SAP Partner Engineering Team at Sun Microsystems India Ltd.

### Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade. SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.