
SRC1.40 Deployment and Sizing Guide

for supported Windows® operating systems

Software version: 1.40 / October 2012

- Documentation Updates..... 2
- In This Version..... 3
 - Service Request Catalog 1.40..... 3
 - Service Manager 9.31 3
- Deployment Guide..... 4
 - Pre-Conditions 4
 - SRC Deployment Models..... 4
 - SRC Standalone Deployment Guide..... 7
 - SRC Horizontal Scaling Deployment Guide 7
- Sizing Guide 13
 - Benchmark Performance Testing 13
 - Sizing Reference Recommended 17
- Support 18
- Legal Notices 19

Documentation Updates

The first page of this release notes document contains the following identifying information:

- Version number, which indicates the software version.
- Publish date, which changes each time the document is updated.

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:
<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:
<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Note: To view files in PDF format (*.pdf), Adobe Acrobat Reader must be installed on your system. To download Adobe Acrobat Reader, go to the following web site:
<http://www.adobe.com/>

In This Version

This document presents how to deploy SRC 1.4 as a standalone model (the simplest model), or as a horizontally-scaled model using Apache HTTP Server to load balance client traffic to different SRC 1.4 instances. It also provides sizing guide for better performance. We recommend reading the document thoroughly to gain a complete sense of the SRC 1.4 deployment, architecture, and how it acts together with Service Manager 9.31.

- [Service Request Catalog 1.40](#)
- [Service Manager 9.31](#)

Service Request Catalog 1.40

Service Request Catalog is a self-service web interface to request IT goods and services, manage those requests, or approve requests for others. This release integrates with Service Manager 9.31, which provides the catalog content, fulfillment, and approval workflow for each request.

Service Manager 9.31

HP Service Manager is a comprehensive and fully integrated IT Service Management software suite that enables IT organizations to improve service levels, balance resources, control costs, and mitigate risk exposure to the organization. It enables you to manage services using a "lifecycle" approach, with consistent improvement built into the governance model.

Deployment Guide

The following section provides an example of how to deploy SRC 1.40 on a Windows Server 2008.

Pre-Conditions

Prepare Server for SRC Deployment

It is recommended to deploy SRC on a Windows Server 2008 separate from that of Service Manager for better performance. Before you start the installation, it is assumed that you have prepared the server and installed the following third-party resources:

- Sun Java JDK 6 or later release
- Apache Tomcat 7

In this document, performance testing for SRC is done on a 64-bit Windows Server 2008. For more details of the configuration, refer to chapter: "[Lab Topologic Configuration](#)".

Service Manager 9.31 Installed

SRC 1.4 works with Service Manager 9.31. Before you start the deployment, you must have an instance of Service Manager 9.31 installed.

Note: If you need help installing Service manager, refer to the Service Manager 9.31 Interactive Installation Guide.

Client Operating System

To access the SRC UI Web service, you must install Adobe Flash Player 10.3 (or later) on client server. For details, refer to the SRC 1.4 Support Matrix.

SRC Deployment Models

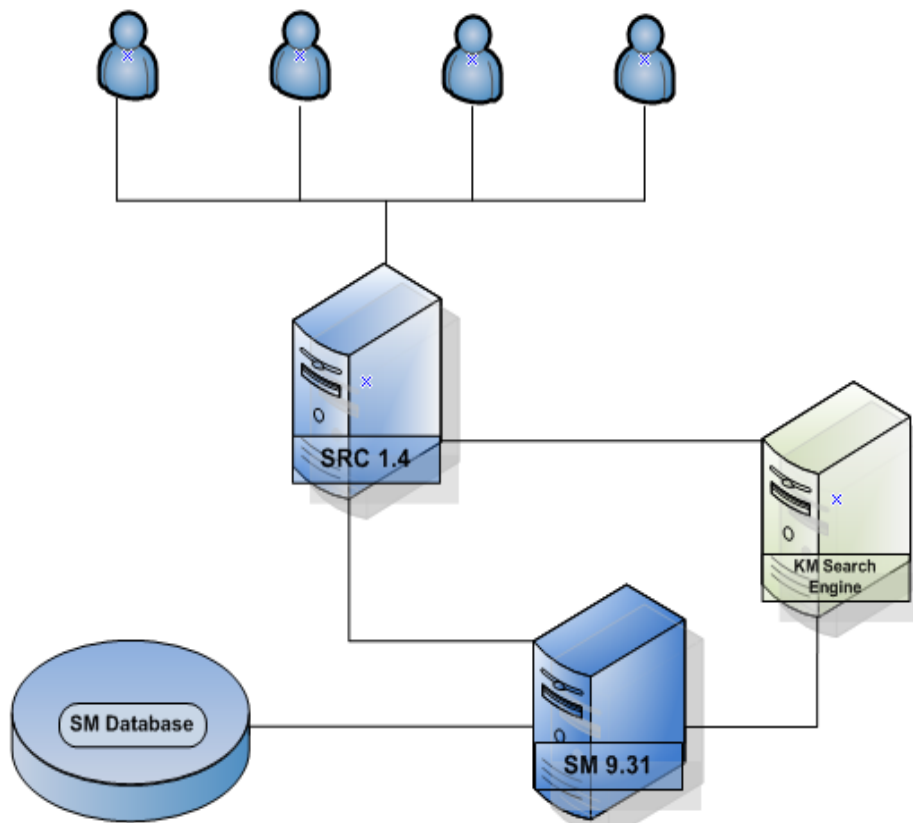
This section will present two different kinds of SRC deployment models, a standalone deployment model and a horizontally-scaled model. The former is the most simplest and basic deployment model, which includes only a single SRC instance. The latter scales multiple SRC instances horizontally to support more users.

Standalone Deployment Model

The architecture shown in Figure 1 is the standalone model, and includes the following components:

1. Service Request Catalog 1.40
2. KM Search Engine
3. Service Manager 9.31
4. Service Manager Dataset

Figure 1 – Standalone Deployment Model



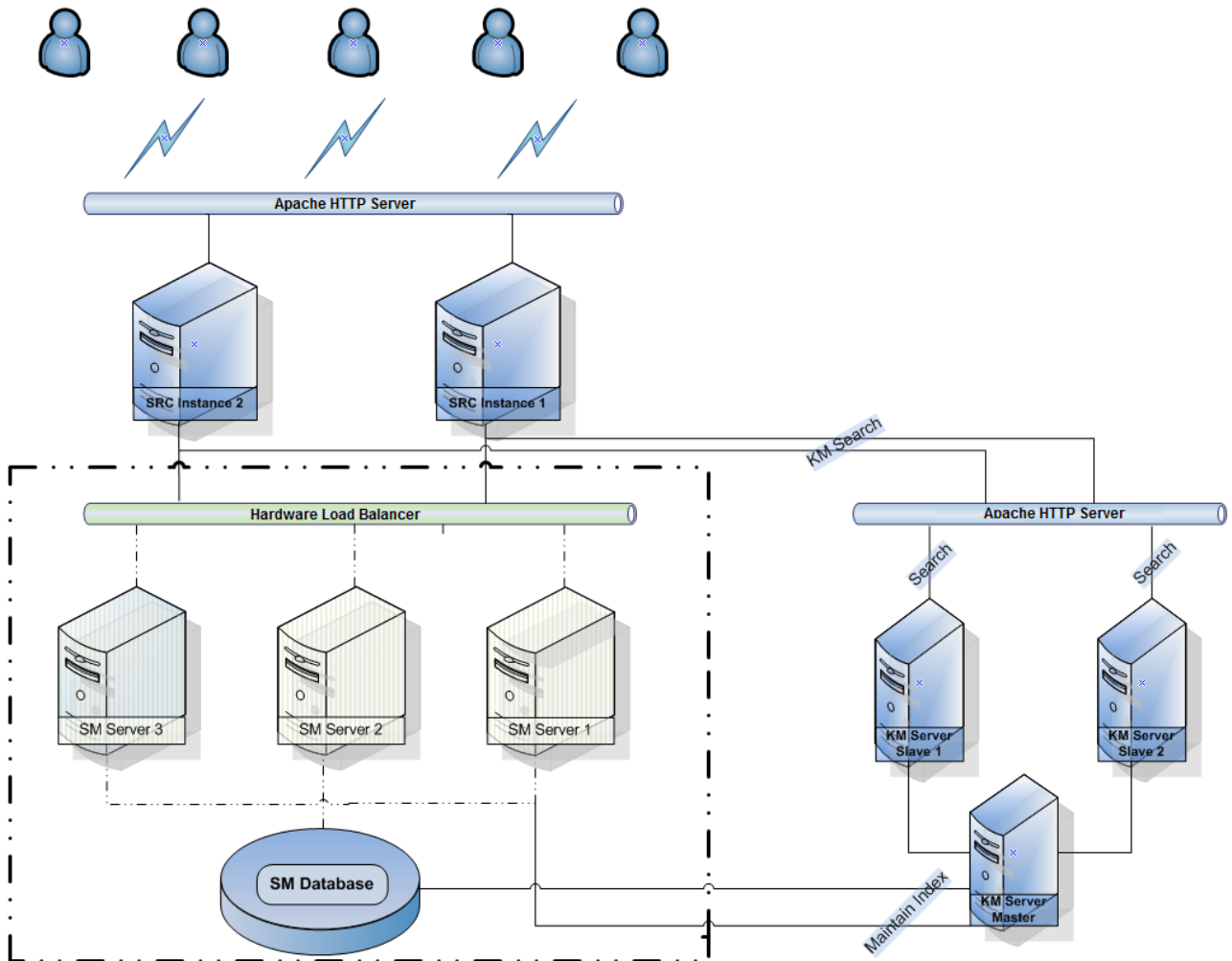
Note: Please refer to [SRC Standalone Deployment Guide](#) for deployment guide.

Horizontal Scaling Model

To support more users, it is recommended to scale multiple SRC applications horizontally, by using an Apache HTTP Server as a software load balancer to direct client traffic to multiple SRC instances. This model also supports hardware load balancer which can direct SRC requests to multiple Service Manager nodes, such as a Cisco ACE 4710 Application Control Engine.

See the following figure for more details on the architecture of this model.

Figure 2 - Horizontal Scaling Model



Note:

1. KM Search Engine supports the Solr search engine. The K2 engine is not supported;
2. KM does not support HTTPS for the Solr search engine;
3. Refer to the Service Manager Search Engine Guide for KM search engine configuration details.
4. The Service Manager deployment is marked by the bold dotted boundary line. For details on this aspect of the figure, refer to the deployment document for Service Manager 9.31.

For the horizontally-scaled model, refer to the [SRC Horizontal Scaling Deployment Guide](#).

SRC Standalone Deployment Guide

Before you deploy SRC, an instance of Service Manager 9.31 must be installed. During the deployment, you will need the Service Manager Port number, and hostname of the server on which Service Manager installed. To finish the SRC standalone deployment, follow these steps:

Deploy and Configure the SRC Instance

This section requires both JDK and Apache Tomcat installed as the preconditions.

Refer to the SRC 1.4 Interactive Installation Guide for the installation details.

KM Search Engine Installation and Configuration

1. For installation of KM, refer to ‘Service Manager 9.31 Knowledge Management Search Engine Guide’.

For configuration of KM, refer to the ‘Enable Knowledge Management’ chapter in the ‘SRC 1.4 Customization Guide’ document.

Verify Your SRC Standalone Deployment

2. Using Internet Explorer, access the SRC UI by specifying a URL in the following format:

`http://<SRC-Server-IP>:<Port>/<path>/secure/main.jsp`

For example: `http://localhost:8080/src-1.40/secure/main.jsp`

3. If successful, you should see the SRC login page. If so, enter “falcon” in the **User Name** field, and then click **Submit** to log in to the SRC system.

SRC Horizontal Scaling Deployment Guide

The architecture of this model refers to Figure 2 in the [Horizontal Scaling Model](#) section.

Topics presented in this chapter are shown below:

1. Deploy and Configure Apache HTTP Server (httpd)
2. Deploy and Configure Multiple SRC Application Instances
3. Verify the Integration

Deploy and Configure Apache HTTP Server (httpd)

Apache HTTP Server can be used as a software load balancer to direct user traffic to SRC instances, as shown in Figure 2 of chapter [Horizontal Scaling Model](#). To deploy using the Horizontal Scaling model, follow these steps:

1. Download the Apache HTTP Server 2.0.64 MSI Installer from following link:
<http://httpd.apache.org/download.cgi>
2. Double-click the MSI Installer to install Apache HTTP Server.
Note: You may change the default installation location. In this document, the installation path is:
`C:\ApacheGroup\`
3. After the installation is complete, specify APACHE_HOME in system variable to
`C:\ApacheGroup\Apache2.`

4. Check whether the mod_jk.so file exists in the <APACHE_HOME>/module directory; If it does not, download this file and place it into the folder<APACHE_HOME>/module. This file is a core component and is the conduit between the Apache HTTP Server and the Tomcat Web application server.
5. To create a workers.properties file in the <APACHE_HOME>/conf directory. The following template is created for multiple Tomcat instances, you may modify this example according to your configuration:

```
worker.list=loadbalancer
worker.tomcat1.port=8001
worker.tomcat1.host=serverHostname
worker.tomcat1.type=ajp13
worker.tomcat1.lbfactor=1

worker.tomcat2.port=8002
worker.tomcat2.host= serverHostname
worker.tomcat2.type=ajp13
worker.tomcat2.lbfactor=1

worker.tomcat3.port=8003
worker.tomcat3.host= serverHostname
worker.tomcat3.type=ajp13
worker.tomcat3.lbfactor=1

.....

worker.tomcatN.port=800N
worker.tomcatN.host= serverHostname
worker.tomcatN.type=ajp13
worker.tomcatN.lbfactor=1

worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=tomcat1, tomcat2,.....,tomcatN
worker.balancer.sticky_session=True
```

Note: Make sure to set sticky_session to **True** to ensure that requests are always routed back to the node which originally assigned the JsessionID.

6. Modify httpd.conf file in ./conf directory according the following example:

```
# WinNT MPM
# ThreadsPerChild: constant number of worker threads in the server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_winnt.c>
ThreadLimit 9000
```



```

ThreadsPerChild 3000
MaxRequestsPerChild 0
</IfModule>
.....
Listen 8080
.....
#ServerName hostname:8080
.....
ServerRoot "C:/ApacheGroup/Apache2"
DocumentRoot "C:/ApacheGroup/Apache2/htdocs"
<Directory "C:/ApacheGroup/Apache2/htdocs">
.....
# We include the /icons/ alias for FancyIndexed directory listings.  If you
# do not use FancyIndexing, you may comment this out.
#
Alias /icons/ "C:/ApacheGroup/Apache2/icons/"
<Directory "C:/ApacheGroup/Apache2/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
#
AliasMatch ^/manual(?:/(?de|en|es|fr|ja|ko|ru|tr))?(/*)?$ "C:/ApacheGroup/Apache2/manual$1"
<Directory "C:/ApacheGroup/Apache2/manual">
    Options Indexes
    AllowOverride None
    Order allow,deny
    Allow from all
    <Files *.html>
        SetHandler type-map
    .....
</Directory>
ScriptAlias /cgi-bin/ "C:/ApacheGroup/Apache2/cgi-bin/"
# "C:/ApacheGroup/Apache2/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
#
<Directory "C:/ApacheGroup/Apache2/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

```

```

.....

# CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
LoadModule jk_module modules/mod_jk.so
JkWorkersFile C:\Apache Group\Apache2\conf\workers.properties
JkLogLevel debug
JkLogFile logs/mod_jk.log

Alias /src-1.40 C:\tomcat1\webapps\src-1.40

JkMount /src-1.40/*.jsp loadbalancer
JkMount /src-1.40/secure/* loadbalancer
JkMount /src-1.40/messagebroker/* loadbalancer
JkMount /src-1.40/configuration/* loadbalancer
JkMount /src-1.40/* loadbalancer

<Location "/src-1.40/WEB-INF/">
AllowOverride None
deny from all
</Location>

```

Note: Make sure all the roots, directories, and alias in httpd.conf file are configured for the correct paths to your Apache HTTP Server installation directory.

After these configuration steps are complete, Apache can function as a software load balancer for multiple SRC instances.

Deploy and Configure Multiple SRC Application Instances

To deploy and configure multiple SRC application instances, follow these steps:

1. Follow the steps in section [Deploy and Configure the SRC Instance](#) to deploy the first SRC instance into Apache Tomcat. For example, specify the path of the first SRC instance as the following:
C:\tomcat1\webapps\src-1.40\
2. Find the server.xml file in the C:\tomcat1\conf directory and adapt the contents as shown in the following example:

Tip 1: change parameter 'Connector prot' to avoid conflicts

```

<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :

Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html

Define a non-SSL HTTP/1.1 Connector on port 8080

```

```
-->
<Connector port="8081" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

Tip 2: definition for AJP 1.3

```
<!-- Define an AJP 1.3 Connector on port 8001 -->
<Connector port="8001" protocol="AJP/1.3" redirectPort="8443" />
```

Tip 3: configure Catalina engine

```
<!-- An Engine represents the entry point (within Catalina) that processes
every request. The Engine implementation for Tomcat stand alone
analyzes the HTTP headers included with the request, and passes them
on to the appropriate Host (virtual host).
Documentation at /docs/config/engine.html -->
<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat1">
```

Tip 4: configure Host and Context

```
<Host name="localhost" appBase="webapps"
unpackWARs="true" autoDeploy="true">
<Context docBase="C:\tomcat1\webapps\src-1.40" path="/src-1.40" reloadable="false"/>
```

3. Modify the Catalina.bat file in the C:\tomcat1\bin directory by setting the CATALINA_HOME parameter to the first Tomcat directory:

```
rem Guess CATALINA_HOME if not defined
set "CURRENT_DIR=%cd%"
set CATALINA_HOME=C:\tomcat1
if not "%CATALINA_HOME%" == "" goto gotHome
set "CATALINA_HOME=%CURRENT_DIR%"
if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
cd ..
set "CATALINA_HOME=%cd%"
cd "%CURRENT_DIR%"
:gotHome
```

4. Deploy the second SRC instance. Copy the tomcat1 folder, where the first SRC instance is deployed to desktop and rename it 'tomcat2'. Then, copy it back to the same directory as tomcat1. Modify the server.xml and catalina.bat files as shown in the following example for the second tomcat instance.

Modify Server.xml for SRC Instance 2:

Step 1: Change 'Connector port' parameter to avoid conflicts

```
<!-- A "Connector" represents an endpoint by which requests are received
and responses are returned. Documentation at :
Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
```

```
Java AJP Connector: /docs/config/ajp.html
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL HTTP/1.1 Connector on port 8080
-->
<Connector port="8082" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

Step 2: Specify the AJP 1.3 protocol

```
<!-- Define an AJP 1.3 Connector on port 8002 -->
<Connector port="8002" protocol="AJP/1.3" redirectPort="8443" />
```

Step 3: Configure the Catalina engine

```
<!-- An Engine represents the entry point (within Catalina) that processes
every request. The Engine implementation for Tomcat stand alone
analyzes the HTTP headers included with the request, and passes them
on to the appropriate Host (virtual host).
Documentation at /docs/config/engine.html -->
<Engine name="Catalina" defaultHost="localhost" jvmRoute="tomcat2">
```

Step 4: Configure the Host and Context

```
<Host name="localhost" appBase="webapps"
unpackWARs="true" autoDeploy="true">
<Context docBase="C:\tomcat2\webapps\src-1.40" path="/src-1.40" reloadable="false"/>
```

Modify the Catalina.bat file for SRC Instance 2:

```
rem Guess CATALINA_HOME if not defined
set "CURRENT_DIR=%cd%"
set CATALINA_HOME=C:\tomcat2
if not "%CATALINA_HOME%" == "" goto gotHome
set "CATALINA_HOME=%CURRENT_DIR%"
if exist "%CATALINA_HOME%\bin\catalina.bat" goto okHome
cd ..
set "CATALINA_HOME=%cd%"
cd "%CURRENT_DIR%"
:gotHome
```

Note: please refer to above configure steps if more SRC instances deployed.

Verify the Integration

After you complete configuration described in Chapters [Deploy and Configure Apache HTTP Server\(httpd\)](#) and [Deploy and Configure Multiple SRC Application Instances](#) , verify whether or not the Apache HTTP Server is working as a software load balancer with multiple SRC instances. To do this, follow these steps:

1. Restart all the SRC instances.

2. Start the Apache HTTP Server. In a Command Prompt window, change to the <APACHE_HOME>/bin directory, and then execute the following command: `apache -k start`

Open Internet Explorer and access the SRC UI Web service by using an appropriate URL, such as the following:

<http://localhost:8080/src-1.40/secure/main.jsp>

Note: In the access.log file, in the ./logs path of the Apache HTTP Server, you may view the details of the request and response status.

Sizing Guide

This section provides sizing guide information to SRC 1.4 customers to achieve a satisfied user experience based on performance testing with 400 concurrent users. The principles in this guideline are based on the critical data that has real impact on Service Request Catalog performance and the tests conducted by HP SRC Performance team.

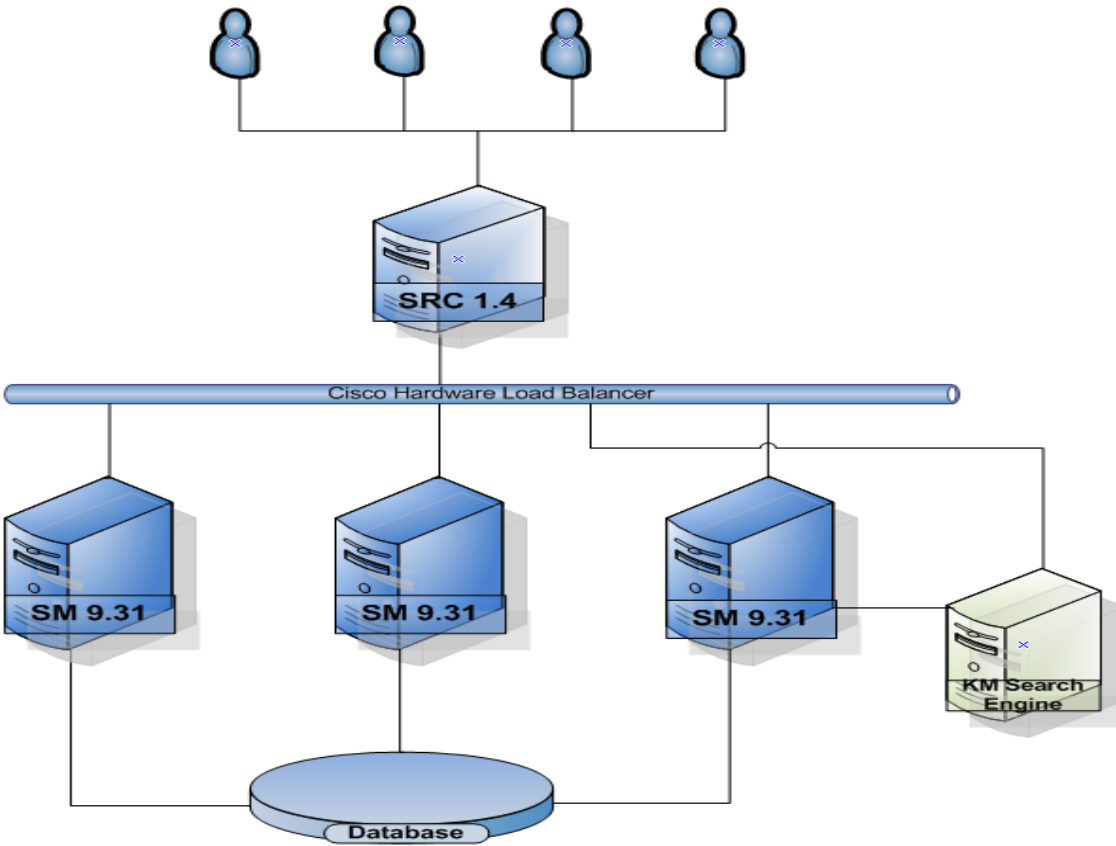
Recommendations in section [Sizing Reference Recommended](#) results from Benchmark Performance Testing, and the data volume in Service Manager Database are 1 million interactions.

Benchmark Performance Testing

- Lab Topologic Configuration

This performance testing is conducted on the following lab topologic architecture: a hardware load balancer balances user requests from the SRC instance to three different Service Manager instances and one KM Search Engine with persistence. The number of nodes that each Service Manager server can support depends on the server configuration.

Figure 3 – Lab Topologic



Note: The Servers used in this performance testing are all physical machines, and the network latency (round trip) is 8ms.

The following table describes the lab configuration:

SRC Server	Configuration
Windows	Windows Server 2008 Enterprise
System	64-bit Operating System
Memory	16.0 G
Processor	Intel® Xeon® CPU X5355 @2.66GHz 2.67GHz (2 processors)
CPU	8 cores
Hard Disk	150 G

SM Server - 1	Configuration
Windows	Windows Server 2008 R2 Enterprise
System	64-bit Operating System
Memory	16.0 G
Processor	Intel® Xeon® CPU X5365 @3.00GHz 3.00GHz (2 processors)

CPU	8 cores
Hard Disk	150 G

SM Server - 2	Configuration
Windows	Windows Server 2008 Enterprise
System	64-bit Operating System
Memory	16.0 G
Processor	Intel® Xeon® CPU X5355 @2.66GHz 2.67GHz (2 processors)
CPU	8 cores
Hard Disk	150 G

SM Server - 3	Configuration
Windows	Windows Server 2003 Enterprise Edition, SP2
System	32-bit Operating System
Memory	64.0 G
Processor	Intel® Xeon® CPU L5430 @2.66GHz 2.67GHz
CPU	8 cores
Hard Disk	150 G

Database Server	Configuration
Windows	Windows Server 2008 Enterprise
System	64-bit Operating System
Memory	64.0 G
Processor	Intel® Xeon® CPU L5430 @2.66GHz 2.67GHz (2 processors)
CPU	8 cores
Hard Disk	150 G

KM Search Engine Server	Configuration
Windows	Windows Server 2003 Enterprise, SP2
System	64-bit Operating System
Memory	16.0 G
Processor	Intel® Xeon® CPU X5355 @2.66GHz 2.67GHz (2 processors)
CPU	8 cores
Hard Disk	150 G

JDK and Apache Tomcat specifications:

1. JDK 1.6.0_31, 64-bit

2. Apache Tomcat 7.0.26, 64-bit

Threads settings in server.xml:

```
maxThreads="600" minSpareThreads="100" maxSpareThreads="300"
acceptCount="575" connectionTimeout="180000" maxKeepAliveRequests="1000"
```

Dataset

Testing is based on 1,000,000 interactions, 6,000 Service Categories, more than 20,000 catalog items, 300,000 users with operator, ESA, ESS roles, and more than 130,000 knowledge management documents in an Oracle 11.2.0 database. The critical data types in regards to Service Request Catalog performance are the service category, catalog items, interactions, km documents and users.

Data Type	Data Volume	Criteria
Service Category	6,000	1. Typical transaction server response time is not slower than 5 seconds.
Service Catalog Item	27,617	
Service Interactions	1,000,000	
km documents	133688	2. Login transaction server response time is not slower than 10 seconds.
km attachments	6088	
Users	5,000 Operator	3. KM search server transaction response time is not slower than 1s.
	5,000 ESA	
	30,000 ESS	

- SRC Configuration Settings

applicationContext.properties	Settings
src.sm.defaultMaxConnectionsPerHost	60
src.sm.maxTotalConnections	60
src.refreshCatalogAfterEvery	36000
src.km.enabled	true
km.defaultMaxConnectionsPerHost	60
km.maxTotalConnections	60
sm.km.doctype.expireDuration	3600

- SM Configuration Settings

sm.ini	Settings
shared_memory	128000000
webservices_sessiontimeout	180
threadperprocess	100
sessiontimeout	10
heartbeatinterval	5

MaxkeepAliveRequests	100
dbcachequery	extaccess
JVMOption0	-Xms512M
JVMOption1	-Xmx512M

Sizing Reference Recommended

- We recommend that the size of the attachment a user uploads while submitting service or support requests is not larger than 30M. Otherwise, users maybe encounter much longer uploading times and/or unexpected exceptions.
- The hard disk on each SRC server should meet the actual requirement of storing items from users. Therefore we recommend a minimum of 60 GB of free space.
- We recommend that you adjust the max threads in server.xml of Apache Tomcat according to the number of potential customers.

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract. To find more information about support access levels, go to the following URL:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to the following URL:

<http://h20229.www2.hp.com/passport-registration.html>

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 1994-2012 Hewlett-Packard Development Company, L.P.

Trademark Notices

- Adobe™ is a trademark of Adobe Systems Incorporated.
- Java™ is a registered trademark of Oracle and/or its affiliates.
- Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.
- Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.
- UNIX® is a registered trademark of The Open Group.

For a complete list of open source and third party acknowledgements, visit the HP Software Support Online web site and search for the product manual called HP Service Manager Open Source and Third Party License Agreements.