



Stephen Wehr

# Linux on IBM zSeries and S/390: z/VM Configuration for WebSphere Deployments

## Abstract

In this Redpaper, we introduce a method to determine an efficient infrastructure for WebSphere deployment on Linux for zSeries. The infrastructure consists of LPARs running multiple Linux guests under VM. Each Linux guest runs WebSphere®, which in turn runs Web applications. In this paper, we describe steps to configure z/VM Linux guests intended to run WebSphere applications. Specifically, we outline a method to calculate the memory required to deploy WebSphere applications, and describe how that memory should be allocated to the LPAR, z/VM®, and Linux guests.

**Note:** This methodology was developed from customer experiences and our own testing. It should work for most customers, but we cannot guarantee that every application fits this model.

## How to use this Redpaper

This paper contains high-level explanations, worksheets, and detailed instructions. We recommend the following process:

1. Read the paper through and fill in the sample worksheets as you go along.
2. Transfer the data from your worksheets to the spreadsheet (available as a download; refer to “System requirements for downloading the Web material” on page 18).
3. Use the data from the spreadsheet to configure the LPARs, VM, and Linux.

## Deployment overview

We discuss a typical WebSphere for Linux for zSeries deployment, shown in Figure 1.

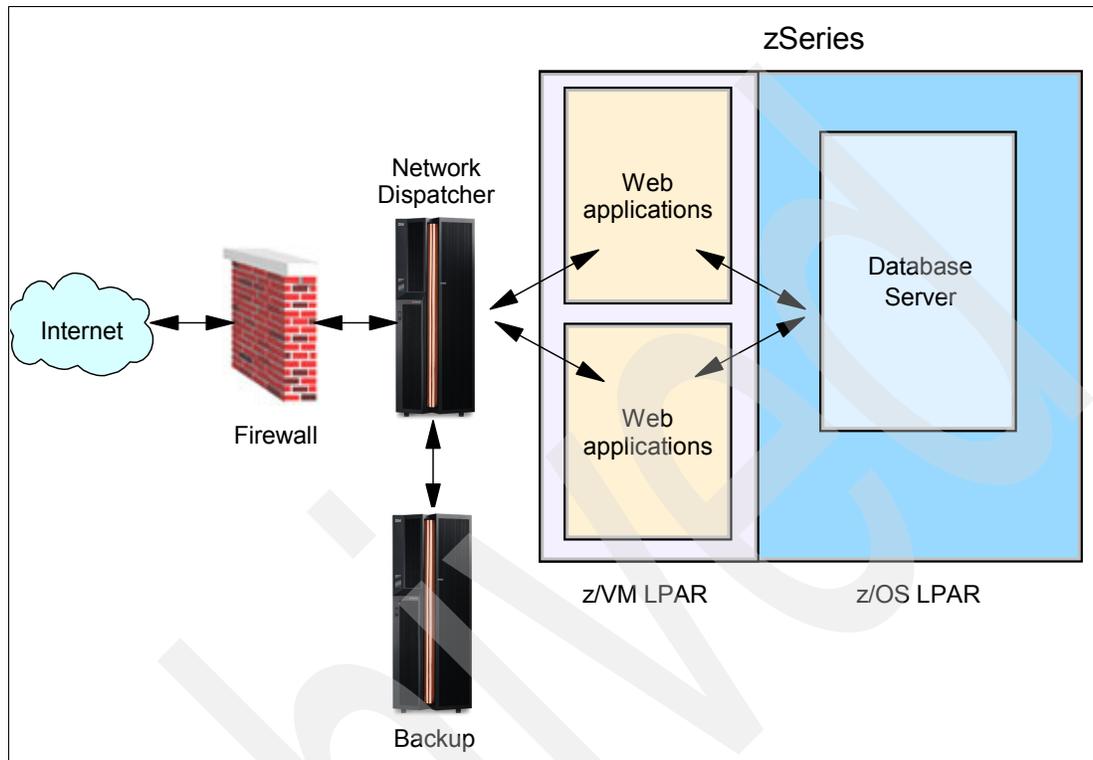


Figure 1 Typical WebSphere deployment on zSeries

In this deployment, a zSeries machine runs one or more z/VM LPARs. WebSphere runs on Linux guests in the z/VM LPAR. The WebSphere applications access DB2® databases running on the z/OS® LPAR using JDBC through DB2 Connect™. A network dispatcher (part of the IBM® Edge Server suite) receives requests from the Internet, and sprays them between two or more production Linux guests running the WebSphere applications.

## Why use z/VM to run Linux

Linux can run natively or it can run under z/VM. When deploying WebSphere, z/VM provides the ability to:

- ▶ Define dozens of Linux servers
- ▶ Quickly create new virtual Linux servers for developers and testers
- ▶ Share CPU and memory between lightly-used Linux servers

For these reasons we recommend using z/VM.

## Steps to deploy the WebSphere applications

We discuss the following steps to deploy the WebSphere applications:

1. "Install the recommended software levels" on page 3
2. "Estimate memory requirements" on page 3
3. "Define and configure the LPARs" on page 4

4. “Configure z/VM” on page 6
5. “Create the Linux guests” on page 8
6. “Install and configure WebSphere” on page 10
7. “Deploy the applications” on page 12
8. “Test the deployment and adjust settings” on page 13

## Install the recommended software levels

Table 1 lists the minimum required software levels.

*Table 1 Minimum software levels for WebSphere deployment*

Component	Minimum recommended level
VM	z/VM 4.3
Linux	SLES 7 (SuSE Linux Enterprise Server) 2.4.7 kernel, with the patch CD applied.
WebSphere	WebSphere Application Server v4.0.2 FixPack5 for Linux for zSeries.
DB2	V7.2. FixPack7

Note that WebSphere supports only applications levels listed in its support matrix found at:

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

**Note:** Concepts discussed in this redpaper are also applicable to z/VM versions 4.1 and 4.2 as well as all levels of WebSphere versions 4 and 5. Be sure to consult the support matrix for the software levels supported by your WebSphere installation.

## Estimate memory requirements

The main task in deploying WebSphere applications on Linux for zSeries is to determine required memory and processor resources. We first estimate application memory requirements. Based on that estimate, we determine:

- ▶ The number of Linux systems (VM guests) needed to run the applications
- ▶ The amount of memory required for each Linux guest
- ▶ The amount of memory to allocate to the VM LPAR

For initial deployment, the development team can provide the best estimate of an application’s memory requirements. If deployed on another platform, an application’s memory requirements may be estimated from experience on that platform. If no better information is available, try an initial estimate of 200 MB for application memory.

### Memory estimate worksheet

Use the worksheet shown in Table 2 on page 4 to estimate memory required to deploy the applications.

Table 2 Worksheet to estimate application memory requirements

Name	Memory estimate		
	Application	Application server	Application total
		60 MB	
	<b>Total</b>		

Add rows as needed for all applications to be deployed. Following is an explanation of each field:

- ▶ **Name**  
The optional name of the application.
- ▶ **Memory estimate**  
Use these columns to estimate memory required for each application:
  - **Application**  
An estimate of required application memory. This is Java Virtual Machine (JVM) size required by the application.
  - **Application server**  
An estimate of the required WebSphere application server memory. Each application server uses memory in addition to the applications running in the application server. The 60MB figure is an average for a typical application server.
  - **Application total**  
Total memory for each application (the sum of **Application** and **Application server**).
- ▶ **Total**  
The total memory required for all applications and applications servers.

## Define and configure the LPARs

Figure 2 on page 5 shows a typical configuration of zSeries® logical partitions (LPARs) for WebSphere applications. A z900 can support up to 16 LPARs.

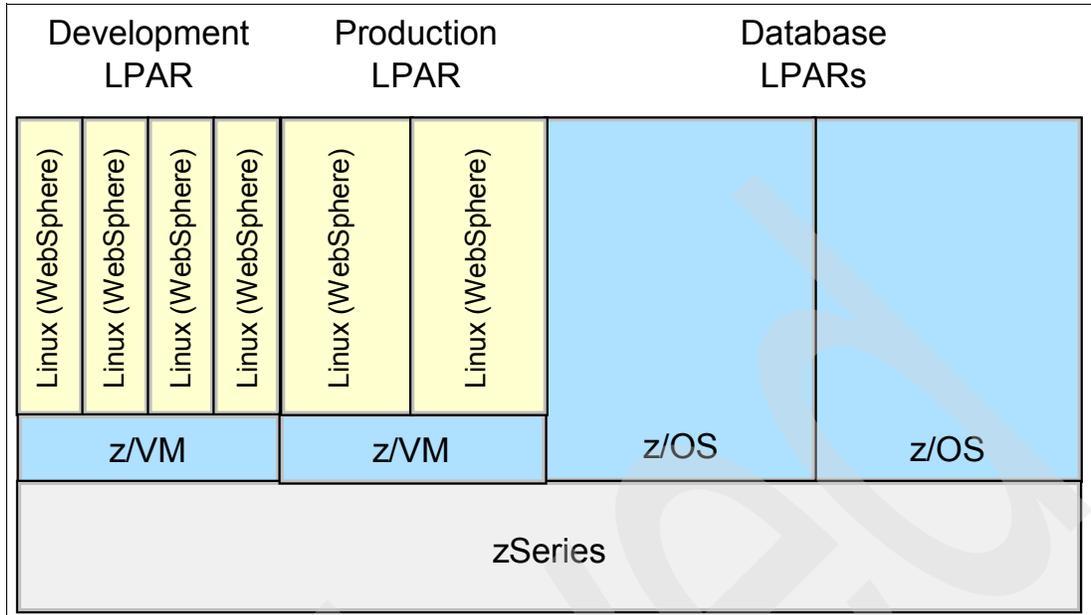


Figure 2 Configuration for WebSphere deployment on zSeries

This configuration consists of two z/VM LPARs:

- ▶ One to run production applications
- ▶ One to develop and test applications before deployment

WebSphere is deployed on z/VM Linux guests in each z/VM LPAR. In general, more Linux guests are defined to the test LPAR. This allows for:

- ▶ **A test environment that duplicates the production environment**  
Before deployment, an application should be tested in a configuration that simulates the production environment. The test LPAR is used to simulate the production LPAR.
- ▶ **Extra Linux guests for developers**  
Define extra Linux guests for developers to use for early stage testing.

### Assigning processor resources to LPARs

Two types of processors are available on zSeries to run your WebSphere applications under Linux: general purpose processors, or Integrated Facility for Linux (IFL) processors:

- ▶ **General purpose CPs**  
These can run all zSeries workloads: z/OS, z/VM, and Linux.
- ▶ **Integrated Facility for Linux (IFL)**  
IFLs use the same hardware as CPs but are limited by microcode to run only z/VM and Linux workloads.

IFLs offer cost savings advantages:

- ▶ IFLs do not add to licensing costs for software running in z/OS LPARs.
- ▶ The purchase price for IFLs is less than general purpose CPs.

However, there is a downside. An LPAR cannot use *both* general purpose CPUs and IFLs; it must use all one type or the other. If IFLs power z/VM LPARs, you lose the freedom to share all CPs across all LPARs. If LPARs configured with CPs have frequent periods of under-utilization, spare CP cycles cannot be used by an LPAR configured with IFLs.

We feel that the advantages of IFLs outweigh the disadvantages and recommend using IFLs to power VM LPARs running Linux and WebSphere.

**Note:** To determine the number of processors required to run the anticipated workload, have your IBM representative contact TechLine (or have your Business Partner contact PartnerLine) and request a sizing from SIZE390. You will be sent a questionnaire that you and your IBM representative can fill out to give TechLine the information they need about your WebSphere applications in order to do an accurate CPU sizing. We strongly recommend using SIZE390.

Although many WebSphere applications can share a single processor, true parallel processing requires at least two real processors. Consider defining at least two processors to the z/VM LPAR to take advantage of real concurrent processing.

In the configuration shown in Figure 2 on page 5, a good approach is to use at least two processors. Each Linux guest in turn can be defined to use two processors. Configure each LPAR to share processors with the following weights:

- ▶ **Production LPAR**  
Assign an LPAR weight of 60%.
- ▶ **Test LPAR**  
Assign an LPAR weight of 40%

The production LPAR is weighted higher to ensure it is allocated sufficient processor resources. Sharing processors provides maximum flexibility in configuring the Linux guests, and maximum use of the resources of your zSeries machine. These weight settings are intended as a starting point; adjustments can be made based on workload.

### Assign memory resources to LPARs

z/VM utilizes two types of memory:

- ▶ **Central storage**  
Often referred to as main memory, central storage is the real memory in which programs execute.
- ▶ **Expanded storage**  
Expanded storage also resides in real memory. However, expanded storage acts as a fast paging device for z/VM; programs do not execute in expanded storage. The amount of expanded storage is configurable in z/VM. Increasing expanded storage reduces real memory available for central storage.

Although it may seem intuitive in a 64-bit architecture to configure all real memory as central storage, we recommend against it. Performance will improve by using expanded storage because z/VM's paging algorithms are tuned to take advantage of expanded storage. We recommend using expanded storage, and configuring real memory in the following proportions:

- ▶ Configure 70% of real memory as central storage.
- ▶ Configure 30% of real memory as expanded storage.

## Configure z/VM

When configuring z/VM, we consider processor and memory recommendations.

## Processor recommendations

When an LPAR is configured for one real processor, z/VM and the Linux guests should be configured for one virtual processor. If multiple real processors are available to the LPAR, Linux guests should also define multiple virtual processors; the number of virtual processors should match the number of real processors. For example, if three real processors are defined to the LPAR, each Linux guest should be defined with three virtual processors.

**Note:** If an LPAR is defined with more than four real processors *and* a Linux kernel at version 2.4.7 or earlier is used, define only up to four virtual processors. Linux kernel version 2.4.17 has been demonstrated to scale well past four processors; if using this kernel or later, define more than four virtual processors.

## VM scheduler resource settings

z/VM typically classifies Linux to be a Q3 virtual machine (a long-running transaction). Changing the default z/VM scheduler settings can improve Linux guest performance. The SRM STORBUF control can reduce the time spent by Linux guests waiting on the eligible list. We recommend setting its value as follows:

```
SET SRM STORBUF 300,200,200
```

Include this setting in the PROFILE EXEC for the operator machine or AUTOLOG1 machine.

## Memory recommendations

Total z/VM memory requirements depend on the memory required for all Linux guests. In “Create the Linux guests” on page 8, we use the application memory estimates discussed in “Memory estimate worksheet” on page 3 to estimate the total required z/VM memory.

When assigning memory to z/VM, it is important to understand that Linux guests share z/VM memory. Virtual machines gain access to memory resources through timeslicing. In z/VM, it is normal for the sum of the virtual machine sizes for all guests to exceed the total available real memory (central storage). The VM scheduler is responsible for ensuring each guest is allocated its fair share of processing time, and to manage memory overcommitment.

For Linux guests running WebSphere applications we recommend overcommitting memory in a 1.5/1 ratio. This means for every 1000 MB of virtual memory needed by a Linux guest, z/VM requires only 666 MB of real memory.

**Note:** This ratio is a rule-of-thumb estimate; adjustments may be required, based on workload.

Allocating the correct amount of DASD paging space to z/VM is critical for performance. z/VM will stop if it runs out of page space. We recommend defining twice as much DASD paging space as total z/VM virtual storage.

The amount of memory required to run z/VM should also be considered. This is all in central storage and is typically around 40 MB.

As an example, if a total of 2700 MB of virtual memory is required for your applications:

- ▶ **Allocate 1840 MB for z/VM memory** ( $40 + (2700 * 0.666 = 1800)$ )  
Configure the total z/VM memory into central storage and expanded storage using the recommended ratio. Remember that the 40 MB of VM memory goes to central storage. For this example, configure:
  - **1300 MB for central storage** ( $(1800 * 0.7) + 40 = 1300$ )

- **540 MB for expanded storage** ( $1800 * 0.3 = 540$ )
- ▶ **Allocate 3680 MB of paging space** ( $1800 * 2 = 3680$ )  
Whenever overcommitting memory, be sure to define enough DASD paging space. Using our recommendation of twice as much paging space as virtual storage, we arrive at the 3680 MB estimate.

The attached spreadsheet will perform these calculations for you.

## Create the Linux guests

A primary consideration when running Linux on z/VM is to reduce guest virtual machine sizes to a minimum. Memory is a shared resource; excessive memory usage by one guest will impact overall system performance. This is particularly important when running Linux guests: Linux attempts to use all its available memory. Memory not used by running applications is allocated to filesystem and I/O buffers by the Linux kernel. On distributed servers with dedicated memory, this policy is beneficial. However, in a shared environment (like VM), aggressive caching can decrease overall performance, because memory held in cache is not available to other guests and can cause excessive VM paging. To overcome Linux's tendency to cache, it is important to keep the size of the Linux virtual machine as small as possible. Linux responds to the smaller virtual machine size by caching less data.

### The 32-bit memory limit

Even though versions the 2.4 Linux kernel and z/VM 4.3 support 64-bit addressing, memory allocation beyond 2 GB will not benefit WebSphere applications. This is due to the fact that current versions of both WebSphere and DB2 support only 32-bit addresses. Do not define virtual machine sizes greater than 2 GB for Linux guests. Instead, if the total memory requirements exceeds 2 GB, use additional Linux guests (keeping the virtual machine size for each below 2 GB).

### A sample calculation

Using the worksheet shown in Table 3 on page 9, we show how to estimate the total memory and number of Linux guests required to deploy five sample applications. (You will use the supplied spreadsheet to do these calculations for your applications.) We begin by using application memory estimates as outlined in "Memory estimate worksheet" on page 3.

Table 3 Sample worksheet to estimate memory requirements

Memory estimate			
Name	Application	Application server	Application Total
App1	500 MB	60 MB	560 MB
App2	300 MB	60 MB	360 MB
App3	550 MB	60 MB	610 MB
App4	200 MB	60 MB	260 MB
App5	500 MB	60 MB	560 MB
<b>Total</b>			2350 MB
<b>Number of Linux guests<sup>a</sup></b>			2
Component	Required memory	Number of Linux guests <sup>b</sup>	Total memory <sup>c</sup>
WebSphere Nanny	25 MB	2	50 MB
WebSphere Admin	100 MB	2	200 MB
Linux and DB2	50 MB	2	100 MB
<b>Component total</b>			350 MB
<b>Total required memory<sup>d</sup></b>			2700 MB

a. Obtained by dividing **Application total** by maximum size of a Linux guest (2 GB, in this case)

b. From **Number of Linux guests**

c. Obtained by multiplying **Required memory** by **Number of guests**

d. Obtained by adding **Application total** and **Component total**

The estimates in the sample show:

- ▶ **2350 MB is estimated for total application memory.**  
Each application runs in its own application server; an additional 60 MB per application is used to account for this expense.
- ▶ **Two Linux guests are required to deploy the applications.**  
Using an upper limit of 2 GB for each Linux guest virtual machine size, two Linux guests are required for the five applications.
- ▶ **The WebSphere nanny process, WebSphere Admin server, DB2, and Linux add additional memory requirements for each guest.**  
Using the estimates for these components, we estimate the total required memory to be 2700 MB.

The choice of applications to deploy on which Linux guest is arbitrary provided the size of any Linux guest does not exceed its maximum (2 GB). For example, The applications could be deployed as follows:

- ▶ **One Linux guest runs App1, App2, App3, and App4; the other guest runs App5.**  
In this case, the first guest requires 1965 MB of memory; the other guest requires 735 MB.
- ▶ **One Linux guest runs App1, App2, and App4; the other guest runs App3 and App5.**  
The first guest requires 1355 MB of memory; the other guest requires 1345 MB.

## Choosing the number of Linux guests

There are advantages in using as few Linux guests as possible (each guest being subject to the 2 GB virtual machine size limit). Total memory can be conserved by sharing the memory used for:

- ▶ **The Linux kernel and DB2**  
Each single Linux guest uses at least 50 MB for the kernel itself and for DB2 (used for the WebSphere repository).
- ▶ **WebSphere**  
The total estimated memory cost of running Websphere is 125 MB (25 MB for the nanny process, 100 MB for the Admin Server). This is in addition to the memory for the application and application server.
- ▶ **Application server**  
Each application server running in WebSphere uses an estimated 60 MB of memory (in addition to application JVM memory).

Additionally, each Linux guest uses processor resources, even when idle. The Linux kernel timer patch greatly reduces this, however some cost still remains.

To minimize memory and processor costs, use as few Linux guests as possible.

## Define a Linux swap device

We recommend using a swap device for each Linux guest. The ideal virtual machine size for each guest is large enough that Linux rarely swaps, but small enough to conserve virtual memory (as outlined in “Memory recommendations” on page 7). We recommend a swap device size approximately 15% of the virtual machine size of the Linux guest. For example, a 1 GB Linux virtual machine should allocate 150 MB for the swap device.

Use a VDISK swap device; it resides in central storage and is much faster than a normal DASD swap device.

## Install and configure WebSphere

The most important WebSphere configuration parameters for performance are the minimum and maximum size of JVM heap. These values are used at initialization and cannot be changed during WebSphere operation (modification requires restarting WebSphere). Factors that require consideration when choosing these values include:

- ▶ If the maximum heap size is reached during execution, an attempt to acquire more JVM heap space results in a program trap.
- ▶ Smaller minimum values result in more frequent attempts to expand JVM heap space. This can lead to excessive Java garbage collection.

These values are specified in the WebSphere Admin Client. Go to **Node** → **YourNode** → **YourApplicationServer** → **Command Line Arguments**. The parameters are **-Xms** (minimum heap size) and **-Xmx** (maximum heap size).

Ideally, both values should be equal and set to the maximum memory required by the application server. If this value has not been quantified, we recommend an initial estimate of 128 MB for the minimum and 400 MB for the maximum. To adjust the values, monitor memory usage at nominal load, then use minimum and maximum settings 20% greater than the observed usage. You can use WebSphere Resource Analyzer to monitor the JVM heap usage.

For more information on the JVM heap size, consult *Performance Analysis for Java Websites* by Stacy Joines, Ruth Willenborg, and Ken Hygh.

## Determining the number of application servers to use

The number of application servers defined in WebSphere affects the total memory requirements. Each application server allocates its own JVM, whether it runs one or many applications. The most efficient use of processors and memory resources is achieved using a single application server running multiple applications. To do this, deploy each application to an existing application server, rather than creating a new application server for the application. This allows memory to be shared by application server and all applications deployed within the application server.

A shared application server saves an estimated 60 MB of memory per application (as opposed to an application server running a single application). Studies indicate that CPU savings also occur when applications share application servers.

However, shared application servers allow poorly behaved applications to adversely affect other applications running in the same application server. Additionally, an update to an application running in a shared application server requires restarting the application server (and therefore, all other applications running in the same application server). Therefore, we recommend:

- ▶ **In the test LPAR, deploy each application in its own application server.**  
This allows applications to be modified, deployed, and tested on a single Linux guest. (The attached spread sheet uses this model.)
- ▶ **Initially deploy each application in its own application server on the production LPAR.**  
This minimizes the potential impact an unstable application may have on a production system.
- ▶ **Once sufficiently tested, move applications in the production LPAR to a shared application server.**  
When an application is demonstrated to be stable, resource savings can be realized by utilizing a shared application server. However, we recommend sharing no more than five applications in an application server.

## Performance tuning for WebSphere

The following recommendations from the Washington Systems Center can improve the performance of your WebSphere applications:

- ▶ **Use the same value for StartServers, MaxClients, and MaxSpareServers parameters in the httpd.conf file.**  
Identically defined values avoid starting additional servers as workload increases. The HTTP server error log displays a message if the value is too low. Use 40 as an initial value.
- ▶ **Serve image content (JPG and GIF files) from the IBM HTTP Server (IHS) or Apache Web server.**  
Do not use the file serving servlet in WebSphere. Use the DocumentRoot and <Directory> directives, or the ALIAS directive to point to the image file directory.
- ▶ **Cache JSPs and Servlets using the servletcache.xml file.**  
A sample definition is provided in the servletcache.sample.xml file. The URI defined in the servletcache.xml must match the URI found in the IHS access log. Look for GET statements, and a definition for each for each JSP or servlet to cache.
- ▶ **Eliminate servlet reloading in production.**  
Specify reloadingEnabled="false" in the ibm-web-ext.xml file located in the application's WEB-INF subdirectory.

- ▶ **Use Resource Analyzer to tune parameter settings.**  
Additionally, examine the access, error, and native logs to verify applications are functioning correctly.
- ▶ **Reduce WebSphere queuing.**  
To avoid flooding WebSphere queues, do not use an excessively large MaxClients value in the httpd.conf file. The Web Container Service General Properties MAXIMUM THREAD SIZE value should be two-thirds the value of MaxClients specified in the httpd.conf file. The Transport MAXIMUM KEEP ALIVE connections should be five more than the MaxClients value.

## Deploy the applications

To summarize, the sample WebSphere deployment is shown in Figure 3.

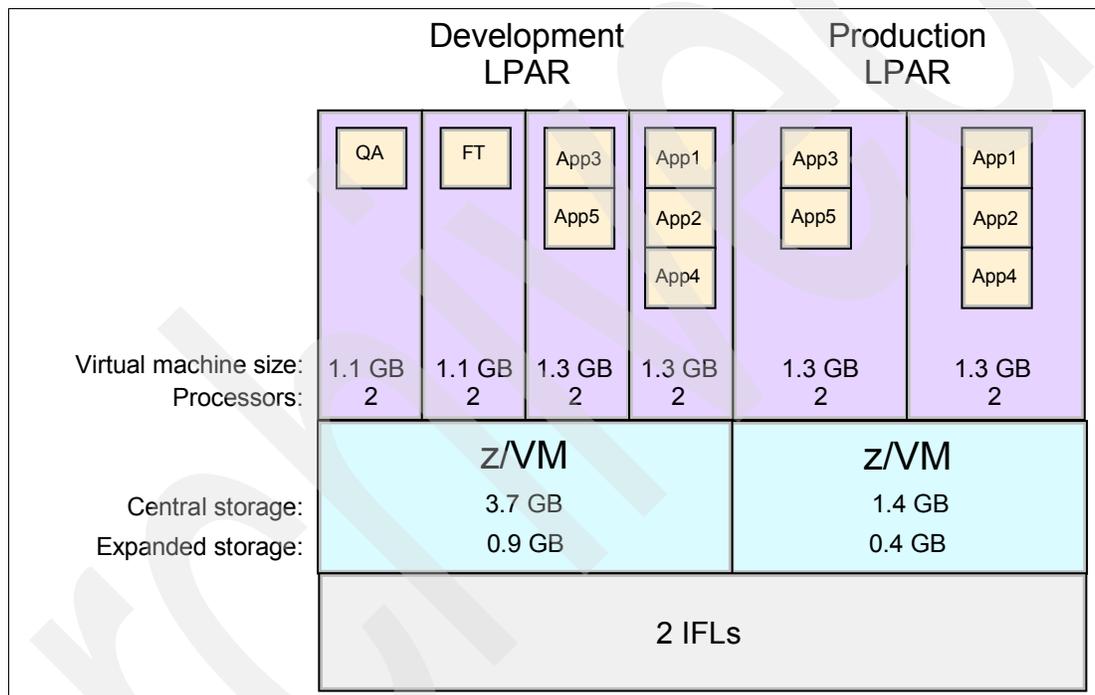


Figure 3 Sample WebSphere deployment on zSeries

### Using the spreadsheet

To assist in obtaining your memory estimates, a Microsoft Excel spreadsheet is available with this redpaper; see “Additional material” on page 17 for instructions on obtaining the spreadsheet. To use the spreadsheet:

1. Enter the memory estimates for each application from Table 3 on page 9 into the spreadsheet. Add these into the **Memory Estimate** column for **Step 1**.

The spreadsheet automatically accounts for application server, WebSphere nanny, and the WebSphere Admin Server, Linux, DB2, and VM memory. These values are shown in the **Assumptions** column in the spreadsheet.

The spreadsheet assumes that each application will run in its own application server.

2. Read the **Results** column to see the calculated number of required Linux guests and total virtual memory.

The swap device size for each Linux guest can be read in the **Swap File** column for the specific guest.

3. The spreadsheet calculates central storage and expanded storage required for z/VM in the **Cstore** and **Estore** columns in **Step 2**.

The **Assumptions** column in **Step 2** shows the assumptions used in making these calculations.

From these calculations, define the z/VM LPARs, the number of Linux guests, and the virtual machine size of each Linux guest.

## Test the deployment and adjust settings

Once deployed, adjustments to the initial memory and processor estimates can be made based on observed performance in a field test environment.

### Adjusting memory

To adjust memory allocation, determine actual application memory usage. Set JVM heap maximum and minimum size to this value plus an additional 20%. To obtain a memory usage estimate:

1. Run the application under moderately heavy load for at least 20 minutes allowing memory usage to peak (assuming no memory leaks).
2. Determine the actual application memory usage.  
Application memory usage can be determined from:
  - The Resident Set Size (RSS) of the Linux processes running the application server. A utility to obtain this value is shown in “The was\_appserver.pl script” on page 14. The RSS can be larger than the JVM storage actually used by the application. RSS size includes JVM memory required for both the application and the application server. Typically, an application server may use 60 MB, but it can require up to 120 MB. Subtracting application server memory from the RSS value yields a rough estimate of actual application memory usage.
  - Profile and monitoring tools.  
Use profiling or monitoring software to obtain a more accurate memory usage estimate. WebSphere Resource Analyzer has the ability to analyze the JVM runtime; this includes total JVM storage. Use the **total**, rather than the **used** memory size. Total memory size indicates current JVM heap size; used memory size indicates memory currently used by the application. Alternatively, turn on verbose garbage collection (**verbosegc**) to see the memory in use by the JVM.

As discussed in “Determining the number of application servers to use” on page 11, shared application servers conserve memory. Adjust the virtual machine size of the Linux guests to reflect observed memory usage. Change memory allocated to the z/VM LPAR as needed. When running the applications, observe z/VM paging. High paging rates indicate additional memory should be allocated to the z/VM LPAR.

### Observing processor usage

If the observed application performance is not acceptable, more processor resource may be required. High processor usage by the WebSphere Linux guest may be an indicator that more processors are required. Follow the recommendation outlined in “Processor recommendations” on page 7 when adding additional processors. Other factors to consider are database performance. Consult *Performance Analysis for Java Websites* by Stacy Joines, Ruth Willenborg, and Ken Hygh for more Java performance related information.

## Monitoring the deployment

Monitoring is an important part of WebSphere deployment. For production, monitoring tools can monitor application health (are applications operating as expected) and performance. During development and test, monitoring can help ensure applications perform as expected and can help identify problem areas. A complete discussion of system monitoring is beyond the scope of this paper; some monitoring tools available for WebSphere on Linux for zSeries include:

- ▶ **IBM Tivoli Monitoring for Web Infrastructure**  
Monitors Edge servers, Web servers, and application servers
- ▶ **IBM Tivoli Web Site Analyzer**  
Monitors Web servers
- ▶ **Wily Technologies Introscope**  
Monitors WebSphere, JVM, application, JDBC, CICS, MQ
- ▶ **IBM WebSphere Studio Application Monitor**  
Monitors WebSphere, JVM, applications, JDBC
- ▶ **IBM WebSphere Resource Analyzer**  
Monitors WebSphere, JVM, applications
- ▶ **Sitraka JProbe**  
Monitors JVM, applications

## Appendix

### The was\_appserver.pl script

This perl script can help determine application memory usage. It displays memory used by WebSphere as well as memory usage for active WebSphere application servers. Using the Linux **ps** command, the script displays all processes containing the text "ActiveEJBServerProcess" (the WebSphere application server process). Using the RSS value for these processes, the script attempts to identify the amount of memory used by WebSphere applications. The script may be obtained as part of the additional material available with this redpaper; see "Additional material" on page 17 for instructions on how to obtain a copy.

Example 1 on page 14 shows how to invoke the utility.

*Example 1 Using the was\_appservers.pl script*

---

```
# ./was_appservers.pl
Displaying WebSphere's Processes and all started WebSphere Application Servers.
Hold on a second while I get the ps output...
Application Server Name          Procs   RSS
-----
WAS Nanny                        7  24580
WAS Admin Server                 49 102712
WebSphere Commerce Payments     44 112696
WebSphere Commerce Server - demo 44 392100
Default Server                   52  87964
-----
#                                720052
```

---

Script output indicates memory used by:

- ▶ WebSphere nanny process
- ▶ WebSphere Admin Server
- ▶ All applications servers

The total RSS memory for all WebSphere components is also displayed. The `was_appserver.pl` script is shown in Example 2.

*Example 2 The was\_appserver.pl script*

---

```
#!/usr/bin/perl
#=====
# program: was_appserver.pl
# Author: Steve Wehr, IBM Corporation
# purpose: Displays all the application servers in WebSphere and the
#          memory used by each
# Syntax: Just invoke it
# Updates: V1.2: Remove VSZ, we're not using it for anything
# Updates: V1.1: Also display the WAS adminserver and nanny processes.
#####
use FileHandle;
$first_time = 1;
$first_as = 1;
# This is the ps command used to get the output
$search_ps = "ps -eo rss,vsize,command";
# This string is used to find Application Servers
$search1 = "ActiveEJBServerProcess";
# This string is used to find the Nanny processes
$search2 = "process.Nanny";
$was_nanny = "WAS Nanny";
# This string is used to find the admin server processes
$search3 = "server.AdminServer";
$was_admin = "WAS Admin Server";
print "Displaying WebSphere's Processes and all started WebSphere Application Servers.\n";
print "Hold on a second while I get the ps output...\n\n";
# Take a snapshot of the ps data and save it to a variable. The ps command
# will display the Resident Set Size, Virtual Size, and full command name
# of each process.
@ps_output = ` $search_ps `;
# Loop through each line of output from ps
LOOP: foreach $line (@ps_output) {
    $_=$line;
    # If the only line returned is our own ps command, then there are
    # no application servers running. Break out of this loop.
    /$search_ps/ && last;
    # Isolate the rss from the rest of the line. It's the first data on the line.
    /^(\\d+) / && ($rss = $1);
    # Isolate the vsize from the rest of the line. It's the second data on the line.
    # VSZ seems useless so I'm not using it anymore, uncomment this if you want to see it.
    #/^(\\d+ (\\d+) / && ($vsize = $1);
    # We are looking for either an:
    # - WAS Admin server process.
    # - WAS Nanny process.
    # - WAS Application server process.
    # Isolate the app server name from the rest of the line.
    if ( /$search1:(.+?):/ ) {
        $process_name = $1;
        $first_as = 0;
    }
    # Isolate the Nanny Process from the rest of the line.
    } elsif ( /$search2/ ) {
        $process_name = $was_nanny;
```



```
STDOUT->format_name("STDOUT_BOT");
write STDOUT;
exit;
```

---

## References

### ITSO Publications

- ▶ *Linux on IBM @server zSeries and S/390: Performance Measurement and Tuning*, SG24-6926  
<http://www.ibm.com/redbooks/abstracts/sg246926.html>
- ▶ *Building Linux Systems Under IBM VM*, REDP0120  
<http://www.ibm.com/redbooks/abstracts/redp0120.html>
- ▶ *Linux for S/390*, SG24-4987  
<http://www.ibm.com/redbooks/abstracts/sg244987.html>
- ▶ *Linux on IBM @server zSeries and S/390: ISP/ASP Solutions*, SG24-6299  
<http://www.ibm.com/redbooks/abstracts/sg246299.html>
- ▶ *Linux on IBM @server zSeries and S/390: Distributions*, SG24-6264  
<http://www.ibm.com/redbooks/abstracts/sg246264.html>

### z/VM documentation

- ▶ *z/VM V4R3 CP Command and Utility Reference*, SC24-6008
- ▶ *z/VM V4R3 CP Planning and Administration*, SC24-6043

### Other resources

- ▶ Stacy Joines, Ruth Willenborg, Ken Hygh, *Performance Analysis for Java Websites*, Addison Wesley Professional, ISBN 0-201-84454-0
- ▶ Tips on configuring z/VM memory  
<http://www.vm.ibm.com/perf/tips/storconf.html>
- ▶ Tips on configuring z/VM Minidisk Cache  
<http://www.vm.ibm.com/perf/tips/prgmdcar.html>

### Referenced Web sites

- ▶ Supported prerequisites for WebSphere Application Server  
<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

### Additional material

This IBM Redpaper refers to additional material which can be downloaded as described below.

## Locating the Web Material

The Web material associated with this Redpaper is available in softcopy on the Internet from IBM Redbooks™ Web Server. Point your browser to:

<ftp://www.redbooks.ibm.com/redbooks/REDP3661/>

Alternatively you can go to the IBM Redbooks site at:

[ibm.com/redbooks](http://ibm.com/redbooks)

Select Additional materials and open the directory that corresponds to Redpaper form number REDP3661.

## Using the Web material

The additional material that accompanies this Redpaper includes:

- ▶ **MemoryEstimating.xls**  
A Microsoft Excel spreadsheet to assist in performing the calculations described in this redpaper.
- ▶ **was\_appserver.pl**  
The Perl script discussed in “The was\_appserver.pl script” on page 14.

## System requirements for downloading the Web material

The following system configuration is recommended:

- Hard disk space:** 30 KB
- Operating System:** Microsoft Windows 98, 2000, ME, or XP
- Applications:** Microsoft Excel. Perl

To install the spreadsheet, download the MemoryEstimating.xls file to a Microsoft Windows workstation. Follow the instructions outlined in “Using the spreadsheet” on page 12.

## About the author

**Stephen Wehr** is a Linux and WebSphere consultant in the zSeries New Technology Center in Poughkeepsie, NY. In his 24 years with IBM, he has held various product development, management, and customer consulting assignments associated with S/390®. For the last three years he has concentrated on helping customers implement WebSphere on Linux for zSeries.

## Acknowledgements

The author would like to thank the following people for their contribution to this Redpaper:

Terry Barthel  
International Technical Support Organization, Poughkeepsie

Richard Lewis  
Washington Systems Center, Gaithersburg

Kevin Curley  
zSeries New Technology Center, Poughkeepsie

## Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an Internet note to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYJ Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

Archived

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

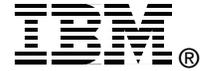
Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on September 11, 2003.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an Internet note to:  
[redbook@us.ibm.com](mailto:redbook@us.ibm.com)
- ▶ Mail your comments to:  
IBM Corporation, International Technical Support Organization  
Dept. HYJ Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400 U.S.A.

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server™  
@server™  
eServer™  
ibm.com®  
CICS®  
DB2 Connect™

DB2®  
IBM®  
OS/390®  
Redbooks™  
Redbooks(logo) ™  
S/390®

Tivoli®  
WebSphere®  
z/OS®  
z/VM®  
zSeries®

The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.