



BEA WebLogic Portal™®

Capacity Planning Guide

Copyright

Copyright © 2004 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Contents

Capacity Planning for WebLogic Portal

Capacity Planning Factors to Consider	1
SSL Connections and Performance	2
WebLogic Server Process Load	3
Database Server Capacity and User Storage Requirements	3
Concurrent Sessions	4
Network Load	4
Clustered Configurations	5
Application Design	6
Tuning Your WebLogic Server	6
Benchmark Data	6
Test Application	6
Test Portals Used	7
Benchmark Results	7
Concurrent Users with Think Time	12
Other Resources	16

Capacity Planning for WebLogic Portal

BEA WebLogic Portal runs on hardware ranging from low-end PCs to high-end mainframes. The process of determining what type of hardware and software configuration is required to meet application needs adequately is called capacity planning.

This document covers the steps involved with capacity planning for WebLogic Portal 8.1 and the application of these techniques by the use of the BEA capacity planning estimation tool.

Capacity planning is not an exact science. Every application is different and every user behavior is different. This document is meant only as a guide for developing capacity planning numbers and will encourage you to err on the side of caution.

Note: Any and all recommendations provided in this guide should be adequately verified before a given system is moved into production. There is no substitute for adequately testing a prototype to obtain capacity planning numbers.

This Capacity Planning Guide contains capacity planning information for WebLogic Portal 8.1. Please contact your BEA sales representative for more information about capacity planning.

Capacity Planning Factors to Consider

A number of factors influence how much capacity a given hardware configuration will need in order to support a WebLogic Portal and a given application. The hardware capacity required to support your application depends on the specifics of the application and configuration. You should consider how each of these factors applies to your configuration and application.

The following sections discuss several of these factors. Understanding these factors and considering the requirements of your application will aid you in generating server hardware requirements for your configuration.

For more information, see the [WebLogic Server Capacity Planning Guide](#).

- Do clients use SSL to connect to WebLogic Portal?
- Does the platform have a WebLogic Server Performance Pack Installed?
- What is running on the machine in addition to WebLogic Portal?
- Is the database a limiting factor? Are there additional user storage requirements?
- How many users need to run simultaneously?
- Is there enough bandwidth?
- Is WebLogic Portal configured for clustering?
- How well-designed is the user application?
- Is WebLogic Portal well-tuned?

SSL Connections and Performance

Secure sockets layer (SSL) is a standard for secure Internet communications. WebLogic Server security services support X.509 digital certificates and access control lists (ACLs) to authenticate participants and manage access to network services. For example, SSL can protect JSP pages listing employee salaries, blocking access to confidential information.

SSL involves intensive computing operations. When supporting the cryptography operations in the SSL protocol, WebLogic Server cannot handle as many simultaneous connections.

You should note the number of SSL connections required out of the total number of clients required. Typically, for every SSL connection that the server can handle, it can handle three non-SSL connections. SSL reduces the capacity of the server by about 33-50% depending upon the strength of encryption used in the SSL connections. Also, the amount of overhead SSL imposes is related to how many client interactions have SSL enabled.

You can also implement SSL using hardware accelerators. Please refer to [WebLogic Server documentation](#).

WebLogic Server Process Load

What is running on the machine in addition to a WebLogic Portal? The machine where a WebLogic Portal is running may be processing much more than presentation and business logic. For example, it could be running a web server or maintaining a remote information feed, such as a stock information feed from a quote service.

Consider how much of your WebLogic Portal machine's processing power is consumed by processes unrelated to WebLogic Portal. In the case in which the WebLogic Portal (or the machine on which it resides) is doing substantial additional work, you need to determine how much processing power will be drained by other processes.

Database Server Capacity and User Storage Requirements

Is the database a bottleneck? Are there additional user storage requirements? Many installations find that their database server runs out of capacity much sooner than the WebLogic Portal does. You must plan for a database that is sufficiently robust to handle the application. Typically, a good application will require a database that is three to four times more powerful than the application server hardware. It is good practice to use a separate machine for your database server.

Generally, you can tell if your database is the bottleneck if you are unable to maintain the WebLogic Portal CPU usage in the 80%-90% range. This is a good indication that your WebLogic Portal is spending much of its time idle and waiting for the database to return. With load balancing in a cluster, the CPU utilization across the nodes should be about even.

Some database vendors are beginning to provide capacity planning information for application servers. Frequently this is a response to the 3-tier model for applications. An application might require user storage for operations that do not interact with a database. For example, in a secure system disk and memory are required to store security information for each user. You should calculate the size required to store one user's information, and multiply by the maximum number of expected users.

Size Estimates Using Oracle

Here are some example Oracle size estimates for RDBMS Authenticator users, based on tablespace changes that take place when creating four million users. This does not include space taken for portal framework customizations. Only the USER_SECURITY and ENTITY tables were tested.

Sizing estimates were estimated for Oracle9i Enterprise Edition Release 9.2.0.5.0.

Average Database Space Needed Per User ID

- WEBLOGIC_DATA tablespace: 140 Bytes/user (134 Mb/million users)
- WEBLOGIC_INDEX tablespace: 207 Bytes/user (198 Mb/million users)

Note: The baseline numbers produced by the benchmark used in this study should not be used to compare WebLogic Portal with other portals or hardware running similar benchmarks. The benchmark methodology and tuning used in this study are unique.

Concurrent Sessions

Determine the maximum number of concurrent user sessions your WebLogic Portal will be called upon to handle. To handle more users, you will need to add more RAM for efficiency. BEA Systems recommends that you install a minimum of 256 MB of memory for each WebLogic Portal instance.

Next, research the maximum number of clients that will make requests at the same time, and how frequently each client will be making a request. The number of user interactions per second with WebLogic Portal represents the total number of interactions that should be handled per second by a given Portal deployment.

Consider also the maximum number of transactions in a given period to handle spikes in demand. For example, in a stock report application, plan for a surge after the stock market opens and before it closes. If your company is broadcasting a Web site as part of an advertisement during the World Series or World Cup Soccer playoffs, you should expect spikes in demand. For benchmark information about concurrent users see [“Concurrent Users with Think Time” on page -12.](#)

Network Load

Is the bandwidth sufficient? Network performance is affected when the supply of resources is unable to keep up with the demand. WebLogic Server requires a large enough bandwidth to handle all of the connections from clients it is to handle. If you are handling only HTTP clients, expect a similar bandwidth requirement as a web server serving static pages.

The primary factor affecting the requirements for a LAN infrastructure is the use of in-memory replication of session information. In a cluster, in-memory replication of session information is the biggest consumer of LAN bandwidth. Consider whether your application will be requiring the replication of session information.

To determine if you do not have enough bandwidth in a given deployment, you should look at the network tools provided by your network operating system vendor. In most cases, including

Windows NT, Windows 2000, and Solaris, you can inspect the load on the network system. If the load is very high, bandwidth may be a bottleneck for your system.

Recommendation

BEA recommends running a gigabit network and a hardware load balancer to optimize network traffic.

Clustered Configurations

Is the WebLogic Portal Server configured to support clusters? Clusters provide session protection and failover via state replication. Customers using clustering should not see any noticeable performance degradation. A number of WebLogic deployments in production involve placing a cluster of WebLogic servers on a single multiprocessor server.

If you are using a web server to forward requests to a WebLogic Server cluster, sometimes the bottleneck can be the web server. This can happen when using the supplied `HttpClusterServlet` and a proxy server, or one of the supported plug-ins. If the response time does not improve after adding servers to the cluster and the web server machine shows a CPU usage over 95%, consider clustering the web server or running the web server on more powerful hardware.

Recommendation

Based on capacity tests with tuned applications, WebLogic Portal is typically CPU-bound. When deciding how much hardware to buy for a production environment, the speed of the processor(s) should be the top priority.

In most cases, WebLogic Server clusters scale best when deployed with one WebLogic Server instance for every two CPUs. However, as with all capacity planning, you should test the actual deployment with your target portal applications to determine the optimal number and distribution of server instances.

Application Design

How well-designed is the application? Badly designed or non-optimized user applications can drastically slow down the performance of a given configuration anywhere from 10% to 50%. The best course is to assume that every application that is developed for WebLogic Portal will not be optimal and will not perform as well as benchmark applications. As a precaution, you should increase the maximum capacity that you calculate or expect.

For more information about designing portals, see the following books:

- [Designing Portals for Optimal Performance](#)
- [WebLogic Portal Performance Tuning Guide](#)

Tuning Your WebLogic Server

Is the WebLogic Portal well-tuned? A WebLogic Server should be tuned using the available tuning guide. If the server is not tuned, expect a decrease in performance.

For more information about tuning WebLogic Server, see [WebLogic Server Performance and Tuning](#).

Benchmark Data

WebLogic Portal ran two types of capacity tests, one to assess throughput (which utilized zero think time), and another to determine the maximum number of concurrent users (which utilized an average think time of 5 seconds).

With both sets of data, a ramp-up style test was used where X number of users were added every Y seconds for an hour, then the test is shut down.

The test was a script that allows each user to log-in and then click through the pages (for all but the very small portal, there were 50 page/book clicks) and then repeat at the first page when the last page is reached. The very small portal has 8 pages, so there were 8 clicks. The test script was run for 60 minutes.

Test Application

The test application is deployed to the cluster as an EAR that contained `.portal` and `.portlet` files. It used form-based authentication so that each user maintained a session during the test. The portals themselves varied in size and portlet type. Each portal tested includes one portlet type including JSP, JSR 168, Pageflow, Struts, Preferences) and the portlets used are considered simple portlets such as “Hello World”-type portlets. Tree optimization was used. No entitlements or customizations were enabled.

Test Portals Used

The portal sizes vary with the following parameters (the number of total portlets in the portal are listed after each one):

- Very Small Portal - 64 portlets

- Small Portal - 500 portlets
- Medium Portal - 1000 portlets
- Large Portal - 2000 portlets
- Very Large Portal - 4000 portlets

With the exception of the very small portal (which has a total of 1 book, 8 pages, and 8 portlets per page) each portal size has a varying number of books (Small - 5, Medium - 10, Large - 20, and Very Large - 40) and in each book are 10 pages with 10 portlets per page.

Benchmark Results

The benchmarks were run on two hardware configurations, HP Linux and Sun Solaris.

Note: The baseline numbers produced by the benchmarks used in this study should not be used to compare WebLogic Portal with other portals or hardware running similar benchmarks. The benchmark methodology and tuning used in this study are unique.

HP Linux Hardware and Server Configurations

The HP Linux test used an eight CPU configuration in which there were four physical machines. Each machine had one running managed server, which translates into one portal and one JVM on each physical machine, for a total of four in the cluster.

- Administration Server: HP ProLiant DL360 G4 -- Dual 3.6 GHz Xeon, 4 GB RAM, 15K RPM SCSI Disks, HyperThreading enabled, RedHat Enterprise Linux AS 3.0 Update 4
- Managed Servers: HP ProLiant DL360 G4 -- Dual 3.6 GHz Xeon, 4 GB RAM, 15K RPM SCSI Disks, HyperThreading enabled, RedHat Enterprise Linux AS 3.0 Update 4
- Database Server: HP ProLiant DL380 G4 -- Dual 3.4 GHz Xeon, 4 GB RAM, 15K RPM SCSI Disks, HyperThreading enabled, Windows 2003 Server Enterprise Edition, Oracle 9.2.0.6
- Load Balancer: F5 Networks Big-IP 1500
- LoadRunner Controller: HP ProLiant DL320 G3 -- 3.6 GHz Pentium 4, 2 GB RAM, 15K SCSI Disk, HyperThreading enabled, Windows 2003 Server Enterprise Edition, LoadRunner 7.8
- JVM: JRockit with `-gc:parallel` setting. 1.5GB of heap was allocated to each server.

HP Linux Results

The servers started with 25 execute threads and the JDBC connection pool was set to start at 25 connections with the ability to grow to 50. The Portal Render Queue was set to 5 (the default). These tests were run with 0 seconds of think time so that the servers would become saturated quickly.

Table 1 JSP Portal Throughput in Pages Per Second

Portal Size	2 CPUs	4 CPUs	6 CPUs	8 CPUs
Very Small	295.2	522.4	775.9	1011.0
Small	251.0	440.9	665.9	862.1
Medium	232.1	418.4	630.0	819.6
Large	205.2	364.0	554.4	723.3
Very Large	162.6	296.0	443.3	570.1

Table 2 JSR 168 Throughput in Pages Per Second

Portal Size	2 CPUs	4 CPUs	6 CPUs	8 CPUs
Very Small	254.4	456.9	687.6	891.9
Small	215.2	392.1	585.7	768.9
Medium	205.6	371.2	555.4	722.8
Large	180.3	330.4	496.6	643.8
Very Large	145.1	266.0	399.8	513.7

Table 3 PageFlow Throughput in Pages Per Second

Portal Size	2 CPUs	4 CPUs	6 CPUs	8 CPUs
Very Small	171.2	269.3	411.9	540.8
Small	123.4	188.0	283.0	375.7
Medium	117.2	185.4	277.6	364.1
Large	108.7	171.7	257.1	338.0
Very Large	92.8	148.4	217.7	285.0

Table 4 Portlet Preferences Throughput in Pages Per Second

Portal Size	2 CPUs	4 CPUs	6 CPUs	8 CPUs
Very Small	328.6	560.9	852.6	1099.2
Small	276.4	486.0	726.7	950.2
Medium	256.1	456.8	686.6	882.8
Large	224.4	391.3	591.1	777.9
Very Large	175.7	314.5	447.6	614.4

Table 5 Struts Throughput in Pages Per Second

Portal Size	2 CPUs	4 CPUs	6 CPUs	8 CPUs
Very Small	258.5	450.1	681.9	875.3
Small	191.5	329.3	496.3	654.2
Medium	183.1	315.8	483.2	620.4

Table 5 Struts Throughput in Pages Per Second (Continued)

Portal Size	2 CPUs	4 CPUs	6 CPUs	8 CPUs
Large	163.8	286.3	433.6	564.8
Very Large	135.1	242.7	360.1	468.2

Sun Solaris Hardware and Server Configurations

The Sun Solaris test used an eight CPU configuration in which there were two physical machines. Each machine had two running managed servers, which translates into two portals and two JVMs on each physical machine, for a total of four managed servers in the cluster.

- Administration Server: Sun Fire v240, 2 x 1.02GHz, 4GB RAM, 10K RPM SCSI Disks, Sun Solaris 9
- Managed Servers: Sun Fire v440, 4 x 1.02GHz, 8GB RAM, 10K RPM SCSI Disks, Sun Solaris 9
- Database Server: HP ProLiant DL380 G4 -- Dual 3.4 GHz Xeon, 4 GB RAM, 15K PRM SCSI Disks, HyperThreading enabled, Windows 2003 Server Enterprise Edition, Oracle 9.2.0.6
- Load Balancer: F5 Networks Big-IP 1500
- LoadRunner Controller: HP ProLiant DL320 G3 -- 3.6 GHz Pentium 4, 2 GB RAM, 15K SCSI Disk, HyperThreading enabled, Windows 2003 Server Enterprise Edition, LoadRunner 7.8
- JVM: Hotspot with `-server` setting. 1.5GB of heap was allocated to each server.

Sun Solaris Results

The servers started with 25 execute threads and the JDBC connection pool was set to start at 25 connections with the ability to grow to 50. The Portal Render Queue was set to 5 (the default). These tests were run with 0 seconds of think time so that the servers would become saturated quickly.

Table 6 JSP Throughput in Pages Per Second, using Solaris

Portal Size	4 CPUs	8 CPUs
Very Small	173.4	298.6
Small	142.7	257.2
Medium	134.0	245.1
Large	109.2	195.1
Very Large	73.9	151.3

Table 7 JSR 168 Throughput in Pages per Second, using Solaris

Portal Size	4 CPUs	8 CPUs
Very Small	143.0	257.7
Small	125.4	217.6
Medium	117.3	217.8
Large	98.6	176.2
Very Large	66.8	150.2

Table 8 Pageflow Throughput in Pages Per Second, using Solaris

Portal Size	4 CPUs	8 CPUs
Very Small	92.1	170.0
Small	70.3	122.8
Medium	69.2	128.3

Table 8 Pageflow Throughput in Pages Per Second, using Solaris

Portal Size	4 CPUs	8 CPUs
Large	64.6	113.2
Very Large	54.4	91.8

Table 9 Portlet Preferences Throughput in Pages Per Second, using Solaris

Portal Size	4 CPUs	8 CPUs
Very Small	194.2	340.0
Small	160.2	265.7
Medium	152.6	252.7
Large	124.6	215.7
Very Large	84.7	181.3

Table 10 Struts Throughput in Pages Per Second, using Solaris

Portal Size	4 CPUs	8 CPUs
Very Small	138.4	223.2
Small	104.8	170.6
Medium	97.9	159.7
Large	79.0	137.5
Very Large	63.0	117.3

Concurrent Users with Think Time

This test established how many concurrent users the test portal could support with a given response time. Goal response times of two seconds and five seconds were used. The number of

concurrent users listed in the table represent the maximum number of running concurrent users under 2 or 5 seconds. One set of data is with the Apache Plug-in for WebLogic Server and the other is with the BigIP F5 hardware load balancer. This test used the HP Linux configuration, see [“HP Linux Hardware and Server Configurations”](#) on page -7.

These tests differed from the benchmark tests in that a think-time was used to mimic a real-world application. The think-time (the time between user requests) was set to 5 seconds with a randomization of +/- 25%.

Apache Results

Table 11 JSP Concurrent Users with Two-Second Response Time Using Apache

Portal Size	4 CPUs	6 CPUs	8 CPUs
Very Small	3113	4718	6220
Small	2545	3769	4599
Medium	2345	3550	4621
Large	2127	3054	4071
Very Large	1536	2438	3284

Table 12 Pageflow Concurrent Users with Two-Second Response Time Using Apache

Portal Size	4 CPUs	6 CPUs	8 CPUs
Very Small	1338	2127	2812
Small	693	925	1311
Medium	616	913	1249
Large	630	786	1157
Very Large	603	798	1024

Table 13 JSP Concurrent Users with Five-Second Response Time Using Apache

Portal Size	4 CPUs	6 CPUs	8 CPUs
Very Small	4351	6446	8653
Small	3439	5149	6432
Medium	3299	4800	5994
Large	2780	4329	5692
Very Large	2231	3250	5046

Table 14 Pageflow Concurrent Users with Five-Second Response Time Using Apache

Portal Size	4 CPUs	6 CPUs	8 CPUs
Very Small	1815	2842	3759
Small	783	1152	1589
Medium	740	1108	1541
Large	724	1100	1478
Very Large	752	911	1309

F5 Networks Big-IP 1500 Results

Table 15 JSP with Two-Second Response Time Using BigIP F5

Portal Size	8 CPUs	10 CPUs	12 CPUs	14 CPUs	16 CPUs
Very Small	5299	6640	7906	9259	10499
Small	4162	5409	6483	7552	8660

Table 15 JSP with Two-Second Response Time Using BigIP F5 (Continued)

Portal Size	8 CPUs	10 CPUs	12 CPUs	14 CPUs	16 CPUs
Medium	4018	5063	6079	7228	8258
Large	3454	4454	5256	6112	7156
Very Large	2678	3368	4223	4773	5495

Table 16 Pageflow with Two-Second Response Time Using BigIP F5

Portal Size	8 CPUs	10 CPUs	12 CPUs	14 CPUs	16 CPUs
Very Small	2508	3206	3876	4455	5140
Small	1274	1420	1680	1885	2405
Medium	1185	1311	1812	1979	2062
Large	1050	1270	1540	1892	1981
Very Large	899	1140	1398	1704	1745

Table 17 JSP with Five-Second Response Time Using BigIP F5

Portal Size	8 CPUs	10 CPUs	12 CPUs	14 CPUs	16 CPUs
Very Small	7366	9237	11250	12963	14760
Small	5944	7468	9001	10576	12037
Medium	5603	7085	8412	9840	11372
Large	4808	6166	7481	8652	9790
Very Large	3780	4765	5695	6677	7691

Table 18 Pageflow with Five-Second Response Time Using BigIP F5

Portal Size	8 CPUs	10 CPUs	12 CPUs	14 CPUs	16 CPUs
Very Small	3400	4450	5074	5878	6980
Small	1568	1737	2174	2364	2885
Medium	1530	1671	2067	2458	2631
Large	1493	1578	2007	2298	2749
Very Large	1250	1437	1798	2031	2307

Other Resources

Remember that WebLogic Portal uses many components from WebLogic Platform. See the following documentation for more information about tuning WebLogic Portal.

- [Designing Portals for Optimal Performance](#)
- [WebLogic Portal Performance Tuning Guide](#)
- [WebLogic Server Performance and Tuning Guide](#)
- [WebLogic Server Capacity Planning Guide](#)
- [Tuning WebLogic JRockit JVM](#)
- BEA's [dev2dev](#) Website