



# BEA

# WebLogic Server

## Capacity Planning Guide

BEA WebLogic Server 6.0  
Document Date: July 2001

## Copyright

Copyright © 2001 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 Manager, BEA WebLogic Collaborate, BEA WebLogic Commerce Server, BEA WebLogic E-Business Platform, BEA WebLogic Enterprise, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Server, E-Business Control Center, How Business Becomes E-Business, Liquid Data, Operating System for the Internet, and Portal FrameWork are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

### **BEA WebLogic Server Capacity Planning Guide**

<b>Part Number</b>	<b>Document Date</b>	<b>Software Version</b>
N/A	July 2001	BEA WebLogic Server Version 6.0

---

# Contents

## 1. Planning for Capacity Requirements

Introduction to Capacity Planning .....	1-1
Capacity Planning Factors to Consider .....	1-2
What Types of Clients Connect to the WebLogic Server Application? ....	1-2
What Types of Traffic Do the Clients Generate?.....	1-3
Does the Platform Have a WebLogic Server Performance Pack Installed? ....	1-3
Do Clients Use SSL to Connect to the WebLogic Server? .....	1-4
How Well-designed is the Application?.....	1-4
Is the Database a Bottleneck?.....	1-4
What Else is Running on the Machine? .....	1-5
Is the Bandwidth Sufficient? .....	1-5
Is the WebLogic Server Tuned?.....	1-6
Is the WebLogic Server Clustered?.....	1-6
How Many Transactions Must Run Concurrently?.....	1-6

## 2. Determining Capacity Requirements

Examining Results from a Baseline Application.....	2-1
Characteristics of the Baseline Application .....	2-2
Windows NT (4.0) Baseline.....	2-3
UNIX (HP-UX 11.0) Baseline .....	2-3
Generating Capacity Planning Requirements.....	2-4
Capacity Planning Guide FAQs .....	2-5
What About Mainframes or Other Types of Hardware? How to Determine Capacity Planning Information for Those systems? .....	2-5
What About Heterogeneous Client Types? For Example, Programmatic AND http Clients? .....	2-5

---

How Do I Get My Application To Run Faster? .....	2-5
What About AIX, Linux, or Other UNIX?.....	2-6
How Do I Determine How Much Memory I Need?.....	2-6
BEA Systems WebLogic Server Documentation .....	A-1
Information from Third-party Software Vendors .....	A-2

---

# About This Document

This document provides guidelines for choosing the appropriate hardware for a customer solution built with the WebLogic Server™ e-commerce platform.

The document is organized as follows:

- Chapter 1, “Planning for Capacity Requirements,” introduces capacity planning and provides information on factors to consider.
- Chapter 2, “Determining Capacity Requirements,” presents a baseline application and guides you through the steps of generating your capacity requirements.

## Audience

This document is written for those invested with responsibilities for capacity planning, and long-range hardware planning for solutions built with the WebLogic Server. It is assumed that readers know Web technologies, object-oriented programming techniques, and client/server computing.

## e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation.

---

# How to Print the Document

You can print a copy of this document from a Web browser, one main topic at a time, by using the File→Print option on your Web browser.

A PDF version of this document is available on the WebLogic Server documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the WebLogic Server documentation Home page, click Download Documentation, and select the document you want to print.

Adobe Acrobat Reader is available at no charge from the Adobe Web site at <http://www.adobe.com>.

## Related Information

References to related information are provided in Appendix A, “Related Reading.”

## Contact Us!

Your feedback on BEA documentation is important to us. Send us e-mail at [docsupport@bea.com](mailto:docsupport@bea.com) if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the documentation.

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at <http://www.bea.com>. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

---

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

## Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Usage
Ctrl+Tab	Keys you press simultaneously.
<i>italics</i>	Emphasis and book titles.
monospace text	Code samples, commands and their options, Java classes, data types, directories, and file names and their extensions. Monospace text also indicates text that you enter from the keyboard.  <i>Examples:</i> <pre>import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float</pre>
<i>monospace</i> <i>italic</i> text	Variables in code.  <i>Example:</i> <pre>String CustomerName;</pre>

---

Convention	Usage
UPPERCASE TEXT	Device names, environment variables, and logical operators. <i>Examples:</i> LPT1 BEA_HOME OR
{ }	A set of choices in a syntax line.
[ ]	Optional items in a syntax line. <i>Example:</i>  <pre>java utils.MulticastTest -n name -a address       [-p portnumber] [-t timeout] [-s send]</pre>
	Separates mutually exclusive choices in a syntax line. <i>Example:</i>  <pre>java weblogic.deploy [list deploy undeploy update]       password {application} {source}</pre>
...	Indicates one of the following in a command line: <ul style="list-style-type: none"> <li>■ An argument can be repeated several times in the command line.</li> <li>■ The statement omits additional optional arguments.</li> <li>■ You can enter additional parameters, values, or other information</li> </ul>
.	Indicates the omission of items from a code example or from a syntax line.

---

# 1 Planning for Capacity Requirements

The following sections provide an introduction to capacity planning for the WebLogic Server e-commerce platform:

- Introduction to Capacity Planning
- Capacity Planning Factors to Consider

## Introduction to Capacity Planning

The BEA WebLogic Server 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 is meant as a guide for Capacity Planning efforts for solutions built with the WebLogic Server.

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

Any and all recommendations generated by this guide should be adequately verified before a given system is put into production. There is no substitute for adequately testing a prototype for capacity planning numbers.

# Capacity Planning Factors to Consider

A number of factors influence how much capacity a given hardware configuration will need in order to support the WebLogic Server and a given application. 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 requirements for your configuration. Consider how much your application differs from the baseline application numbers provided in “Examining Results from a Baseline Application.”

## What Types of Clients Connect to the WebLogic Server Application?

WebLogic Server uses T3, an optimized protocol, to transport data to other Java programs, including clients and other WebLogic Servers. WebLogic Server keeps track of every Java Virtual Machine (JVM) with which it connects, and creates a single T3 connection to carry all traffic for a JVM.

For example, if a Java client accesses an enterprise bean and a JDBC connection pool on WebLogic Server, a single network connection is established between the WebLogic Server JVM and the client JVM. The EJB and JDBC services can be written as if they had sole use of a dedicated network connection because the T3 protocol invisibly multiplexes packets on the single connection.

T3 is an efficient protocol for Java-to-Java applications because it avoids unnecessary network connection events and uses fewer OS resources. The protocol also has internal enhancements that minimize packet sizes

There are two typical types of clients that can connect to the WebLogic Server: programmatic clients (Java applications, applets, etc.) and web-based clients (web browsers, http proxies, etc.). Typically, programmatic (such as Java applications and applets) clients connect through T3 and use RMI to connect to the server. Web clients use the http protocol to get HTML or servlet output.

Programmatic clients are generally more efficient than http clients because T3 does more of the presentation work on the client side. Programmatic clients typically call directly into EJBs while web clients usually go through servlets. This eliminates the work the server must do for presentation. Also, the stateless nature of `http://` requires that the server handle more in terms of overhead. The T3 protocol works over sockets and has a long-standing connection to the server. However, the benefits of http clients are numerous, such as ubiquity of browsers and firewall compatibility, and are usually worth the performance costs.

A WebLogic Server installation that relies only on programmatic clients should be able to handle 20% more concurrent clients than an `http://` which is serving installations. If you are tunneling T3 over http, you should not see this performance benefit. In fact, performance of T3 over http is typically 15% worse than typical http and will reduce the optimum capacity of your WebLogic Server installation similarly.

## **What Types of Traffic Do the Clients Generate?**

If you are using T3 clients, there are a number of ways that interaction with the server is accomplished. Most cases involve either RMI or events. In the case of RMI, clients typically do not generate a large amount of traffic to the server because there is only one sender and one listener. In the case of Events however, one single event with many listeners can generate a large amount of traffic for the server.

Events versus RMI changes the capacity of the server slightly depending upon the architecture of the application. In most cases, this means only that you will need to have a larger bandwidth available between clients and servers.

## **Does the Platform Have a WebLogic Server Performance Pack Installed?**

If you install and implement a Performance Pack for your platform, you should expect a 50% increase in the number of simultaneous connections that the WebLogic Server installation can handle compared to the baseline numbers in “Examining Results from a Baseline Application.”

# Do Clients Use SSL to Connect to the WebLogic Server?

SSL is a very intensive computing operation. Cryptography operations in the SSL protocol mean that the WebLogic Server will not be able to 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. So, SSL reduces the capacity of the server by 33-50% depending upon the strength of encryption used in the SSL connections.

Beginning with version 5.1, WebLogic Server includes native performance packs for SSL operations.

# How Well-designed is the Application?

In the end, the WebLogic Server simply is a platform for user applications. Badly designed or unoptimized 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 the WebLogic Server will not be optimal and will not perform as well as benchmark applications. As a precaution, you should double the maximum capacity that you expect.

There are a number of resources available in the WebLogic Server documentation and the Customer Support centers that are available to help in designing the most efficient applications possible on WebLogic. For the latest information, please visit the WebLogic Server web sites.

# Is the Database a Bottleneck?

The vast majority of installations find that their database server runs out of capacity much sooner than the WebLogic Server 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.

You can generally tell if your database is the bottleneck if you are unable to get the WebLogic Server CPU usage in the 85%-95% range. This is a good indication that your WebLogic Server 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.

## **What Else is Running on the Machine?**

In many cases, the machine running the WebLogic Server is doing other things besides presentation and business logic. For example, the application may be tasked with running a Web server as well. Or, it could be charged with maintaining a remote information feed, such as a stock information feed from a quote service.

In the case in which the WebLogic Server (or the machine on which it resides) is doing substantial additional work, you need to determine how much processing power will be relegated to other processes. In the cases of a web server proxy running on the same machine as the WebLogic Server, you should expect anywhere from 25%-50% of the computing capacity to be relegated to processes that are not related to WebLogic applications compared to the baseline numbers in “Examining Results from a Baseline Application.”

## **Is the Bandwidth Sufficient?**

It is important that a WebLogic Server have a large enough bandwidth to handle all of the connections from clients it is supposed to handle. In the case of programmatic clients, each client JVM will have a single socket to the server. Each socket requires bandwidth. A WebLogic Server handling programmatic clients should have 125%-150% the bandwidth that a similar web server would handle. If you are interested in the bandwidth required to run a web server, you can assume that each 56k/s of bandwidth can handle between seven and ten simultaneous requests depending upon the size of the content that you are delivering.

If you are handling only http clients, expect a similar bandwidth requirement as a web server serving static pages.

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 and Solaris, you can look at what the load is on the network system. If the load is very high, bandwidth may be a bottleneck for your system.

## **Is the WebLogic Server Tuned?**

The WebLogic Server should be tuned using the available tuning guide. If the server is not tuned, expect a 33% decrease in performance compared to the baseline numbers in “Examining Results from a Baseline Application.”

## **Is the WebLogic Server Clustered?**

Clusters are much more efficient and better for failover. Customers using clustering should not see any noticeable performance degradation by doing so. A number of WebLogic deployments in production involve placing a cluster of WebLogic servers on a single multiprocessor server.

## **How Many Transactions Must Run Concurrently?**

Analysis of this factor requires that you complete some calculations. Determine how many sessions your WebLogic Server will be expected to handle concurrently. For example, your requirement may be to have 1000 simultaneous client-server sessions. For each of these sessions, you will need to have more RAM to keep things running smoothly. Obtain as much RAM as you can. BEA Systems recommends that you install a minimum of 256 MB of memory for each WebLogic Server that will be handling more than minimal capacity.

Next, research the maximum number of clients that will make requests at the same time. In the current scenario, there are 1000 simultaneous connections. Let us project that we expect users to make new requests to the server every 20 seconds. That means, on average, the WebLogic Server will need to handle 50 operations in a second to keep up. So, it might appear that we need a machine that can do 50 transactions per second (tps).

Not exactly. You must look at the maximum number of transactions in a given period to handle spikes in demand. In this case, let us assume a surge in demand and each client is making a request every 10 seconds. That means the system must be able to handle 100 tps.

Now that you have considered factors that affect capacity, the next steps are to examine baseline results from a sample application, and to generate your own capacity planning requirements.

# **1** *Planning for Capacity Requirements*

---

# 2 Determining Capacity Requirements

The following sections provide information on how to determine your capacity requirements:

- Examining Results from a Baseline Application
- Generating Capacity Planning Requirements
- Capacity Planning Guide FAQs

## Examining Results from a Baseline Application

WebLogic Server development has produced some baseline capacity planning results as a starting point. These baseline numbers were generated using a typical customer application from the online banking industry. You can compare the parameters of the baseline configurations to your configuration, and use this information to help estimate your capacity requirements.

# Characteristics of the Baseline Application

This application implements standard operations from the banking industry including entity beans for database information, and JSPs (servlets) for presentation. This application was designed to be representative of an actual customer application, but by design, no real optimization has been done on the application itself. This qualifies this application as a “poorly designed” application.

This list summarizes the characteristics of the baseline application:

- SSL is not used.
- Servlets call the EJBs.
- Reasonably inefficient EJB design.
- Clients all make requests through `http://`
- WebLogic Servers are not clustered.
- WebLogic Servers are tuned.
- Performance pack is installed.
- Database is not the bottleneck.

BEA Systems ran the application on the WebLogic Server on two types of machines: Windows NT and HP-UX. If you are looking for capacity planning guidelines for Windows NT, use the Windows NT numbers as your baseline. If you are running on Solaris or another UNIX, use HP-UX as your baseline.

**Note:** Solaris and other UNIX platform numbers should be similar to those on HP-UX.

**Note:** THE BASELINE NUMBERS LISTED HERE ARE NOT BENCHMARKS, AND THEY ARE NOT COUNTS OF TRANSACTIONS PER SECOND. THEY SHOULD NOT BE CONFUSED WITH COMPETITIVE BENCHMARKS. The baseline numbers are conservative baselines based on a poorly designed WebLogic Application on a poorly configured WebLogic Server. A number of benchmarks show how a well-designed application can perform on WebLogic Server. These benchmarks are available from BEA Systems. For more information, contact your BEA Systems sales representative.

## Windows NT (4.0) Baseline

For Windows NT, tests were run using a 1-4way Windows NT 4.0 machine with 4 PIII 450Mhz Xeon chips as the server, and standard PC machines as clients.

**Table 2-1 Windows NT Transactions per Second**

<b>CPUs, each 450 mhz</b>	<b>Memory</b>	<b>Percentage of Baseline Number</b>
1 CPU, Multiprocessor	1 GB	100
2 CPUs, Multiprocessor	1 GB	151
4 CPUs, Multiprocessor	1 GB	204

**Note:** THE BASELINE NUMBERS LISTED HERE ARE NOT BENCHMARKS AND THEY ARE NOT A COUNT OF TRANSACTIONS PER SECOND. DO NOT USE THEM AS COMPETITIVE BENCHMARKS. The baseline numbers are conservative, baseline numbers based on a poorly designed WebLogic Server application on a poorly configured WebLogic Server. A number of benchmarks show how a well-designed application can perform on WebLogic. These benchmarks are available from BEA Systems. For more information, contact your BEA Systems sales representative.

## UNIX (HP-UX 11.0) Baseline

For UNIX (HP-UX 11.0), tests were run using a 1-6way HP-UX 11 machine with 1-6 PA-RISC chips and standard PC machines as clients.

**Table 2-2 HP-UX Transactions per Second**

<b>CPUs</b>	<b>Memory</b>	<b>Percentage of Baseline Number</b>
1 CPU, Single Processor Machine 160 mhz	1 GB	100
2 CPUs, Multiprocessor Machine 240 mhz	1.5 GB	306

CPUs	Memory	Percentage of Baseline Number
4 CPUs, Multiprocessor Machine 240 mhz	2 GB	483
6 CPUs, Multiprocessor Machine 240 mhz	2 GB	600

# Generating Capacity Planning Requirements

Reprising our 1000-user example from “How Many Transactions Must Run Concurrently?” the configuration is running 1000 simultaneous users. Given the details that we found in this example, you can conclude that it will scale similarly to the baseline numbers. This example does not differentiate substantially from the baseline configuration. Therefore, a cluster of 2 4-CPU Windows NT boxes or a 6-CPU box will likely do the job.

For your application, take the information that you know from “Examining Results from a Baseline Application” to see how your application differs from the baseline application. If, for example, the only difference is that you are doing all SSL connections and running on Windows NT, you should expect that you will see 66% less than with the baseline numbers. Do that same logical process for all of the factors listed in “Capacity Planning Factors to Consider.”

Lastly, remember that 99.9% of applications are not well implemented or designed. Take the hardware configuration that you developed in the previous paragraph and double it.

Now, you should be ready to deploy your application with a good idea as to system capacity. Remember! TEST the capacity of your system BEFORE relying on it for production deployments.

# Capacity Planning Guide FAQs

This section is a summary of FAQs related to information in this guide.

## **What About Mainframes or Other Types of Hardware? How to Determine Capacity Planning Information for Those systems?**

It should be relatively simple. Look at the mainframe as a multiple of many individual PCs. For example an E10000 from Sun is a 64-way machine. A conservative estimate would be to look at that machine as being as powerful as 64 individual PCs.

## **What About Heterogeneous Client Types? For Example, Programmatic AND http Clients?**

When in doubt, assume the worst. In this case, assume that 100% of the clients will be http clients and develop your capacity planning numbers appropriately.

## **How Do I Get My Application To Run Faster?**

You will want to tune your WebLogic Server first, using the tuning guide included in BEA Systems documentation. Second, consider using a tuning tool such as jProbe from sitraka, formerly KL Group, at <http://www.sitraka.com> to find bottlenecks in the code. Also, check the latest in the WebLogic Server documentation and customer support FAQs to see up-to-date application tuning and design guides.

### **What About AIX, Linux, or Other UNIX?**

You can expect similar performance across the multiple UNIX-type variants available. However, you should note that some of the UNIX-type operating systems supported by WebLogic Server do not have performance packs.

### **How Do I Determine How Much Memory I Need?**

BEA Systems recommends that you install a minimum of 256 MB of memory for each WebLogic Server that will be handling more than minimal capacity. If you are expecting a very heavy load, BEA Systems recommends that you increase your memory substantially.

For information on memory requirements, see the [BEA WebLogic Server Performance and Tuning Guide](http://e-docs.bea.com/wls/docs60/perform/index.html) at <http://e-docs.bea.com/wls/docs60/perform/index.html>.

# A Related Reading

This section provides references to topics related to capacity planning, including the following:

- BEA Systems WebLogic Server Documentation
- Information from Third-party Software Vendors

## BEA Systems WebLogic Server Documentation

The BEA corporate Web site provides all documentation for WebLogic Server. Information related to capacity planning is available in these topics:

- [Preparing to Install WebLogic Server: System Requirements at http://e-docs.bea.com/wls/docs60/install/instpre.html#1043851](http://e-docs.bea.com/wls/docs60/install/instpre.html#1043851)
- [BEA WebLogic Server Performance and Tuning at http://e-docs.bea.com/wls/docs60/perform/index.html](http://e-docs.bea.com/wls/docs60/perform/index.html)
- [BEA WebLogic Server Performance and Tuning, Performance Analysis Tools at http://e-docs.bea.com/wls/docs60/perform/appa\\_reading.html#1027508](http://e-docs.bea.com/wls/docs60/perform/appa_reading.html#1027508)
- [BEA WebLogic Server Performance and Tuning, Benchmarking Tools at http://e-docs.bea.com/wls/docs60/perform/appa\\_reading.html#1027508](http://e-docs.bea.com/wls/docs60/perform/appa_reading.html#1027508)

# Information from Third-party Software Vendors

Information on topics related to capacity planning is available from numerous third-party software vendors, including the following:

- Standard Performance Evaluation Corporation at <http://www.spec.org>
- jProbe from sitraka, formerly KL Group at <http://www.klgroup.com>
- iPlanet Sun-Netscape Alliance at [http://www.iplanet.com/about\\_us/about\\_iplanet/about\\_iplanet\\_5\\_0a.html](http://www.iplanet.com/about_us/about_iplanet/about_iplanet_5_0a.html)