



Sun Java™ System

Application Server 7

企业版入门指南

2004Q2

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

文件号码: 817-6877-10

版权所有 © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

本文档所介绍产品中涉及的技术的相关知识产权归 Sun Microsystems, Inc. 所有。特别是（但不仅限于），这些知识产权可能包含 <http://www.sun.com/patents> 上列出的一项或多项美国专利以及在美国和其它国家/地区的一项或多项其它专利或正在申请的专利。

本产品包含 Sun Microsystems, Inc. 的保密信息和商业秘密。未经 Sun Microsystems, Inc. 事先明确书面许可，禁止使用、泄露或复制本产品。

美国政府权限 - 商业软件。政府用户必须遵守 Sun Microsystems, Inc. 标准许可证协议的条款以及 FAR 及其补充内容中的适用条款。

使用本软件必须遵守许可证条款的规定。本软件可能包括由第三方开发的产品。

Sun、Sun Microsystems、Sun 徽标、Java、Solaris、Sun™ ONE、Sun™ ONE Studio、iPlanet、J2EE、J2SE、Enterprise JavaBeans、EJB、JavaServer Pages、JSP、JDBC、JDK、JVM、Java Naming and Directory Interface、JavaMail 和 Java 咖啡杯徽标是 Sun Microsystems, Inc. 在美国和其它国家/地区的商标或注册商标。

所有 SPARC 商标的使用均已获得许可，它们是 SPARC International Inc. 在美国和其它国家/地区的商标或注册商标。带有 SPARC 商标的产品均基于 Sun Microsystems, Inc. 开发的体系结构。

UNIX 是由 X/Open Company, Ltd. 在美国和其它国家/地区独家许可的注册商标。

本产品受美国出口控制法控制，并遵守其它国家/地区的进出口法律。严禁将本软件直接或间接用于核武器、导弹、生化武器或核潜艇的研制或使用。严禁出口或转口到美国禁运的国家/地区或美国禁止出口清单中的实体，包括但不限于被禁止的个人和特别指定的国家/地区清单。

本文档按“原样”提供，对所有明示或暗示的条件、陈述和担保，包括对适销性、适用性和非侵权性的暗示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。

目录

- 关于本指南 7**
 - 本指南的内容 7
 - 使用文档 8
 - 本指南的组织方式 10
 - 文档惯例 10
 - 一般惯例 11
 - 有关目录的惯例 12
 - 联系 Sun 12
 - 向我们提供反馈信息 12
 - 获得培训 12
 - 与产品支持中心联系 13
- 第 1 章 Sun Java System Application Server 7 企业版简介 15**
 - 关于 Sun Java System Application Server 15
 - 关于应用服务器实例 16
 - 关于管理域 18
 - 关于企业版功能 18
 - 群集 18
 - 高可用性 19
 - 可扩展性 20
 - 负载平衡 20
 - 会话持久性 22
 - 关于会话持久性类型 22
 - 关于会话持久性配置 24
 - 关于单一登录会话信息 25
 - 关于 SFSB 检查点 25
 - 具有高可用性的数据库 26

配置和管理应用服务器的工具	26
---------------------	----

第 2 章 群集方案	29
-------------------------	-----------

HTTP 群集方案	29
-----------------	----

RMI/IIOP 群集方案	31
---------------------	----

服务器端配置	32
--------------	----

客户端配置	33
-------------	----

独立的客户机	33
--------------	----

ACC 客户机	34
---------------	----

第 3 章 企业功能配置教程	35
-----------------------------	-----------

使用教程之前的准备工作	35
-------------------	----

安装需求	36
------------	----

使用教程之前需要执行的步骤	36
---------------------	----

教程步骤概述	37
--------------	----

启动服务器	39
-------------	----

设置 PATH 变量	39
------------------	----

运行 <code>asadmin start-domain</code>	40
--	----

检验服务器的启动	41
----------------	----

检验管理服务器的启动	41
------------------	----

访问管理界面	42
--------------	----

查看管理服务器的日志	43
------------------	----

检验服务器实例的启动	44
------------------	----

使用 <code>asadmin</code> 检验实例	44
------------------------------------	----

通过访问 HTTP 服务器来检验实例	44
--------------------------	----

创建 <code>loadbalancer.xml</code> 文件	45
---	----

将群集添加到 <code>loadbalancer.xml</code> 文件	47
---	----

配置负载均衡	49
--------------	----

配置运行状况检查器	49
-----------------	----

启用负载均衡器的监视功能	51
--------------------	----

负载均衡器的其它特性	52
------------------	----

使用重新装入轮询时间间隔来进行动态重新配置	52
-----------------------------	----

响应超时	53
------------	----

HTTPS 路由选择	53
------------------	----

样例 <code>loadbalancer.xml</code> 文件	53
---	----

第 4 章 Cluster JSP 样例应用程序教程	55
---	-----------

使用 Cluster JSP 样例应用程序教程的准备工作	56
------------------------------------	----

将 Cluster JSP 样例应用程序部署到群集中	56
----------------------------------	----

<code>cladmin</code> 命令的输入文件	58
------------------------------------	----

<code>cladmin</code> 命令的语法	58
----------------------------------	----

运行 <code>cladmin deploy</code> 命令	59
---	----

由 <code>cladmin</code> 命令支持的 <code>asadmin</code> 命令	60
要求和限制	60
使用 <code>cladmin</code> 命令启动应用服务器实例	61
验证应用程序部署	61
监视应用服务器中的样例	63
使用管理界面查看日志	63
使用 <code>tail</code> 命令查看日志	64
事件日志中由应用程序生成的消息	64
访问日志中由应用程序生成的消息	65
应用服务器实例部署的错误诊断	65
将样例应用程序添加到群集	66
应用配置更改并重新启动 Web 服务器	67
运行应用程序	68
验证 HTTP 负载均衡	73
验证负载均衡的步骤	73
对负载均衡器插件进行错误诊断	74
查找 <code>loadbalancer.xml</code> 文件中的错误	75
使用运行状况检查器	76
验证 HTTP 会话持久性	77
停顿服务器实例	78
 第 5 章 摘要和后续步骤	81
 索引	83

关于本指南

本前言将介绍 《Sun Java™ System Application Server 7 入门指南》的内容。

前言中介绍了以下主题：

- [本指南的内容](#)
- [使用文档](#)
- [本指南的组织方式](#)
- [文档惯例](#)
- [联系 Sun](#)

本指南的内容

本入门指南适用于初次使用 Sun Java System Application Server 的用户。本指南简要介绍了熟悉 Sun Java System Application Server 的企业版功能的实际操作方法，并着重演示了负载平衡功能和 HTTP 会话持久性功能。尽管本指南中的练习并不需要预先具备应用服务器使用和开发经验，但在开始练习之前，必须根据 《*Sun Java System Application Server 安装指南*》安装服务器。

本指南首先向您介绍 Sun Java System Application Server，包括对群集、负载平衡和会话持久性功能的简要概述。接下来，本指南将按步骤指导您配置环境，以便运行练习中的样例应用程序。本指南将向您展示如何部署样例应用程序，以便演示和验证负载平衡和会话持久性。同时还提供一个列表，其中包括深入研究的领域和相应的参考信息。

使用文档

Sun Java System Application Server 标准版和企业版手册的形式为可移植文档格式 (PDF) 或超文本标记语言 (HTML) 格式的联机文件。

下表列出了 Sun Java System Application Server 手册中所介绍的任务和概念。标有（已针对 7 2004Q2 版进行了更新）的手册已针对 Sun Java System Application Server 标准版和企业版的 7 2004Q2 版进行了更新。没有此标记的手册自发布第 7 版企业版至今一直没有更新。

表 1 Sun Java System Application Server 文档指南

有关信息	请参见
（已针对 7 2004Q2 版进行了更新）有关软件和文档的最新信息。其中包括以表格形式对所支持的硬件、操作系统、JDK 和 JDBC/RDBMS 所做的全面概述。	发行说明
Sun Java System Application Server 7 概述，包括每个版本的产品所具有的功能。	<i>Product Overview</i>
Sun Java System Application Server 体系结构的示意图和说明，以及此体系结构方法的优点。	<i>Server Architecture</i>
Sun Java System Application Server 7 中新增的企业功能、开发者功能和操作功能。	<i>What's New</i>
如何开始使用 Sun Java System Application Server 产品，包括样例应用程序教程。	入门指南
（已针对 7 2004Q2 版进行了更新）安装 Sun Java System Application Server 标准版和企业版软件及其组件，例如样例应用程序和管理界面。对于企业版软件，提供了实现高可用性配置的指导。	安装指南
（已针对 7 2004Q2 版进行了更新）评估系统需求和企业状况，确保以最适合您的站点的方式部署 Sun Java System Application Server。此外还介绍了部署应用服务器时应该注意的常见问题。	<i>System Deployment Guide</i>
应用程序架构者和开发者使用 HTTP 会话可用性的最佳做法。	<i>Application Design Guidelines for Storing Session State</i>
创建和实现要在 Sun Java System Application Server 上运行的 Java™ 2 平台企业版（J2EE™ 平台）应用程序，这些应用程序遵循用于 J2EE 组件，例如 Servlet、企业 JavaBean™ (EJB™) 和 JavaServer Page™ (JSP™)，的开放式 Java 标准模型。包括有关应用程序设计、开发者工具、安全性、汇编、部署、调试和创建生命周期模块的常规信息。此外还包括 Sun Java System Application Server 综合词汇表。	<i>Developer's Guide</i>
在 Sun Java System Application Server 上创建和实现 J2EE Web 应用程序，这些应用程序遵循 Java™ Servlet 和 JavaServer Page (JSP) 规范。介绍了 Web 应用程序编程的概念和任务，并提供了样例代码、实现提示和参考资料。其中包括以下主题：结果缓存、JSP 预编译、会话管理、安全性、部署、SHTML 和 CGI。	<i>Developer's Guide to Web Applications</i>
（已针对 7 2004Q2 版进行了更新）在 Sun Java System Application Server 上创建和实现 J2EE 应用程序，这些应用程序遵循企业 Bean 的开放式 Java 标准模型。介绍了企业 JavaBean (EJB) 编程的概念和任务，并提供了样例代码、实现提示和参考资料。其中包括以下主题：容器管理的持久性、只读 Bean 以及与企业 Bean 关联的 XML 和 DTD 文件。	<i>Developer's Guide to Enterprise JavaBeans Technology</i>

表 1 Sun Java System Application Server 文档指南（续）

有关信息	请参见
（已针对 7 2004Q2 版进行了更新）在 Sun Java System Application Server 上创建可访问 J2EE 应用程序的应用程序客户机容器 (ACC)。	<i>Developer's Guide to Clients</i>
在 Sun Java System Application Server 环境中创建 Web 服务。	<i>Developer's Guide to Web Services</i>
Java™ 数据库连接 (JDBC™)、事务、Java Naming and Directory Interface™ (JNDI)、Java™ 消息服务 (JMS) 和 JavaMail™ API。	<i>Developer's Guide to J2EE Services and APIs</i>
创建自定义 NSAPI 插件。	<i>Developer's Guide to NSAPI</i>
（已针对 7 2004Q2 版进行了更新）有关从管理界面和命令行界面配置、管理和部署 Sun Java System Application Server 子系统和组件的信息和指导。其中包括以下主题：群集管理、高可用性数据库、负载平衡和会话持久性。此外还包括 Sun Java System Application Server 综合词汇表。	管理指南
编辑 Sun Java System Application Server 配置文件，例如 <code>server.xml</code> 文件。	<i>Administrator's Configuration File Reference</i>
配置和管理 Sun Java System Application Server 操作环境的安全性。其中包括有关一般安全性、证书和 SSL/TLS 加密的信息。同时还介绍了基于 HTTP 服务器的安全性。	<i>Administrator's Guide to Security</i>
为 Sun Java System Application Server 配置和管理 J2EE™ 连接器体系结构 (CA) 连接器的服务提供者实现。其中包括以下主题：管理工具、池监视器、部署 JCA 连接器，以及连接器样例和应用程序样例。	<i>J2EE CA Service Provider Implementation Administrator's Guide</i>
（已针对 7 2004Q2 版进行了更新）将应用程序迁移到新的 Sun Java System Application Server 编程模型，特别是从 iPlanet Application Server 6.x 和 Sun ONE Application Server 7.0 进行迁移。其中包含一个迁移样例。	<i>Migrating and Redeploying Server Applications Guide</i>
（已针对 7 2004Q2 版进行了更新）调整 Sun Java System Application Server 来提高性能的方法和原因。	<i>Performance Tuning Guide</i>
（已针对 7 2004Q2 版进行了更新）有关解决 Sun Java System Application Server 问题的信息。	<i>Troubleshooting Guide</i>
（已针对 7 2004Q2 版进行了更新）有关解决 Sun Java System Application Server 错误消息的信息。	<i>Error Message Reference</i>
（已针对 7 2004Q2 版进行了更新）适用于 Sun Java System Application Server 的实用程序命令（以手册页的样式编写）。	<i>Utility Reference Manual</i>
使用 Sun Java System Message Queue 3.5 SP1 软件。	Sun Java System Message Queue 文档，位于： http://docs.sun.com/db?p=prod/s1.s1msgqu

本指南的组织方式

本指南为那些初次使用 Sun Java System Application Server 的用户提供了 Sun Java System Application Server 的企业版功能的概述。

本指南包括以下几章：

- [第 1 章 “Sun Java System Application Server 7 企业版简介”](#) 提供了产品概述，其中包括对群集、负载平衡、HTTP 会话持久性和高可用性数据库的简要介绍。
- [第 3 章 “企业功能配置教程”](#) 包含开始教程前的准备任务的列表，配置服务器以使用企业版功能所要执行的步骤汇总，以及有关配置服务器以使用群集和 HTTP 会话故障切换的教程。
- [第 2 章 “群集方案”](#) 包含配置 HTTP 和 RMI/IIOP 的负载平衡和故障切换的操作步骤。
- [第 4 章 “Cluster JSP 样例应用程序教程”](#) 介绍 Cluster JSP 应用程序，以及将其部署到群集中和运行该应用程序的方法。还介绍了使用该应用程序来验证负载平衡和 HTTP 会话持久性的方法。
- [第 5 章 “摘要和后续步骤”](#) 介绍了在本指南中完成的任务，以及使用本产品可进行的后续步骤。

在本指南结尾处，提供了 [“索引”](#)。

文档惯例

本部分介绍在本指南中使用的各种惯例：

- [一般惯例](#)
- [有关目录的惯例](#)

一般惯例

本指南中使用了以下一般惯例：

- 文件和目录路径采用 UNIX® 格式（用正斜杠分隔目录名称）。对于 Windows 版本，目录路径是相同的，但用反斜杠分隔目录。

- URL 的格式如下：

<http://server.domain/path/file.html>

在这些 URL 中，*server* 是运行应用程序的服务器的名称；*domain* 是您的 Internet 域名；*path* 是服务器的目录结构；*file* 是单个文件名。URL 中的斜体项为占位符。

- 字体惯例包括：
 - monospace 字体用于样例代码和代码列表、API 和语言元素（例如，函数名和类名）、文件名、路径名、目录名称以及 HTML 标记。
 - 斜体样式用于代码变量。
 - 斜体样式还用于书名、强调、变量、占位符以及斜体文字。
 - 粗体样式用于段落标题或粗体文字。
- 本文档中，多数平台的安装根目录都由 *install_dir* 表示。例外情况在[第 12 页上的“有关目录的惯例”](#)中进行了说明。

默认情况下，多数平台上的 *install_dir* 都位于：

- 基于文件的 Solaris 和 Linux 安装：

用户主目录 /sun/appserver7

- 所有 Windows 安装：

system drive:\Sun\AppServer7

对于以上列出的平台，*default_config_dir* 和 *install_config_dir* 与 *install_dir* 相同。有关例外情况及其它信息，请参见[第 12 页上的“有关目录的惯例”](#)。

- 本文档中，实例根目录由 *instance_dir* 表示，它是以下内容的缩写：

default_config_dir/domains/domain/instance

- 除非特别说明，否则本手册中所有特定于 UNIX 的说明也适用于 Linux 操作系统。

有关目录的惯例

默认情况下，使用基于软件包的 Solaris 安装或基于 RPM 的 Linux 安装时，应用服务器文件分布在多个根目录中。对应于所提供的各个不同的默认安装目录，本指南使用了以下文档惯例：

- *install_dir* 为 /opt/SUNWappserver7，它包含安装映像的静态部分。组成应用服务器的所有公用程序、可执行文件和库均驻留在此目录中。
- *default_config_dir* 为 /var/opt/SUNWappserver7/domains，它是所创建的域的默认位置。
- *install_config_dir* 为 /etc/opt/SUNWappserver7/config，它包含与安装有关的配置信息，例如，为此安装配置的许可证和管理域主列表。

注意 本手册中的所有 Forte for Java 均已更名为 Sun Java Studio。

联系 Sun

您可能会出于以下目的希望联系 Sun Microsystems：

- [向我们提供反馈信息](#)
- [获得培训](#)
- [与产品支持中心联系](#)

向我们提供反馈信息

如果是关于产品或文档的一般反馈，请将其发送至 appserver-feedback@sun.com。

获得培训

可从以下位置学习 Application Server 培训课程：

http://training.sun.com/US/catalog/enterprise/web_application.html/

经常访问该站点可学习有关 Sun Java System Application Server 的新课程。

与产品支持中心联系

如果您在使用系统时遇到问题，请用以下方式之一联系客户支持：

- 联机支持 Web 站点位于：
<http://www.sun.com/supporttraining/>

- 与维护合同相关的电话号码

在与客户支持联系之前请提供以下信息。这样有助于确保我们的支持人员更好地帮助您解决问题：

- 问题的说明，包括问题发生时出现的情况及其对操作的影响
- 计算机类型、操作系统版本，以及产品版本，包括任何可能对问题产生影响的修补程序和其它软件。以下是一些常用的命令：
 - **Solaris:** pkginfo、showrev
 - **Linux:** rpm
 - **所有:** asadmin version --verbose
- 操作的详细步骤，以便再现问题
- 任何错误日志或信息转储
- 配置文件，例如：
 - `instance_dir/config/server.xml`
 - Web 应用程序的 `web.xml` 文件（如果问题涉及到 Web 应用程序）
- 问题是应用程序在群集中运行时出现的还是单独运行时出现的

联系 Sun

Sun Java System Application Server 7 企业版简介

本章包括以下数节：

- [关于 Sun Java System Application Server](#)
- [关于企业版功能](#)
- [配置和管理应用服务器的工具](#)

关于 Sun Java System Application Server

Sun Java™ System Application Server 7 企业版提供了一个高性能的 Java™ 2 Platform, Enterprise Edition (J2EE™) 平台，可广泛适用于各种应用服务和 Web 服务的部署。本产品提供了一个基于行业标准组件的全新模块式体系结构，包括经过认可的 HTTP 服务器基础架构的实现、Java 消息服务 (JMS)，以及对最新的 J2EE 和 Web 服务规范的严格支持，这些规范适用于 Java Web Services Developer Pack 软件中的 J2EE 1.3 版、Java 2 Platform, Standard Edition 1.4 版和用于 XML 的 Java API (JAX)。

此外，Sun Java System Application Server 还提供了 HTTP/S 和 RMI/IIOP 请求的负载平衡功能以及远程 EJB 引用和 HTTP 会话的高可用性。有关这些功能的完整列表，请参见 *Sun Java System Application Server Product Introduction*。

本节包括以下数个主题：

- [关于应用服务器实例](#)
- [关于管理域](#)

关于应用服务器实例

应用服务器实例构成了应用服务器部署的基础。每个实例都有自己的目录结构、配置和已部署的应用程序。每个服务器实例还包含了 J2EE 平台的 Web 和 EJB 容器。Sun Java System Application Server 还包含一个经过认可的高性能 HTTP 服务器。对象请求代理 (ORB) 模块允许使用 RMI-IIOP 进行 EJB 调用。

为了访问后端系统，应用程序可以利用 J2EE 连接器体系结构支持和第三方资源适配器（内置 JMS 供应程序或第三方供应程序），以及若干种常用的第三方 JDBC 驱动程序中的任意组合。对后端系统的访问可以使用内置的纯 Java 语言编写的事务管理器在分布式事务范围内进行管理。

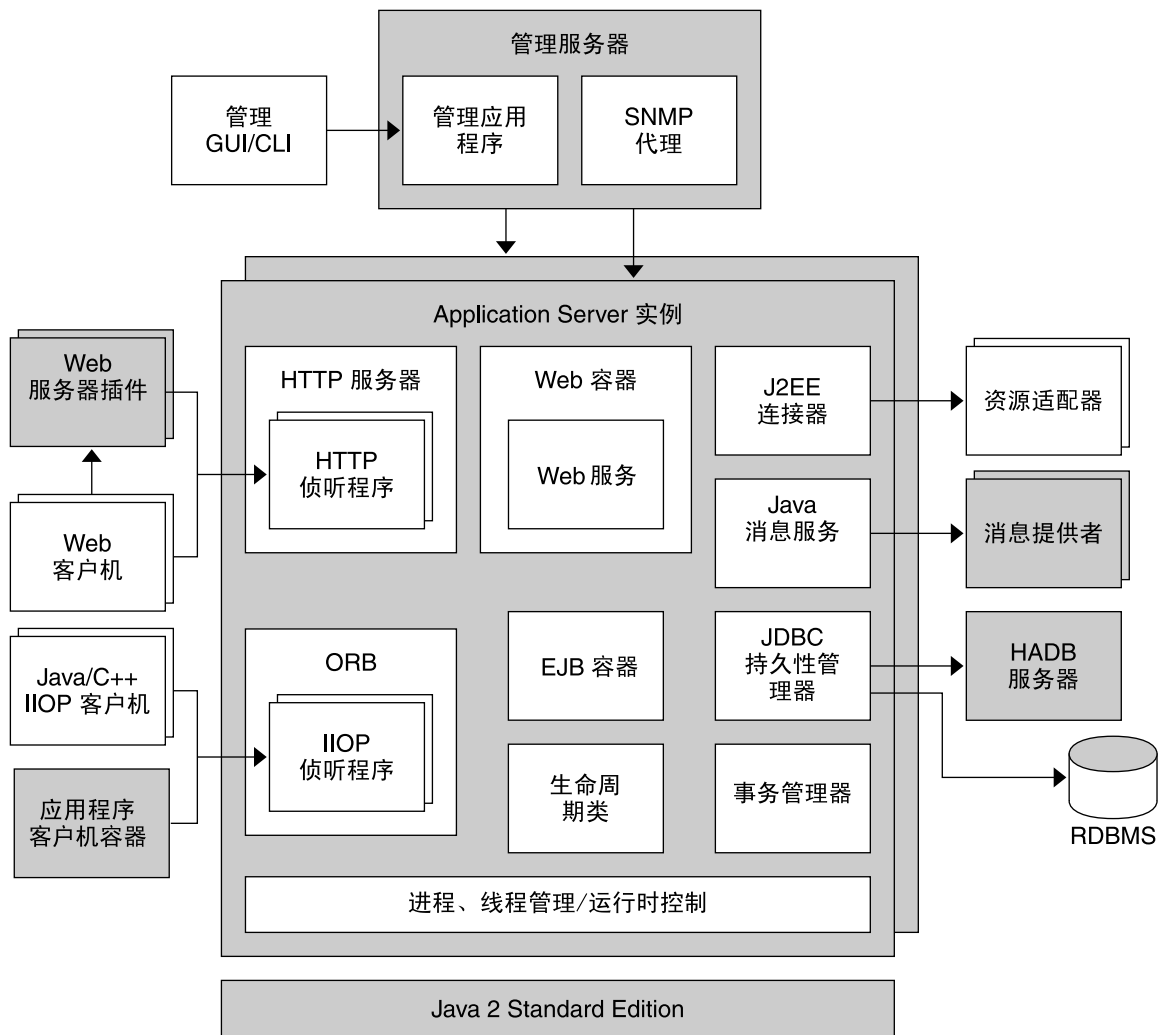
管理服务器是一个特殊的服务器实例，它运行着核心管理应用程序和一个 SNMP 代理。所有的远程管理活动均流经管理服务器。基于命令行和基于 Web 浏览器的管理客户机均通过 HTTP 或 HTTPS（较为安全）直接访问管理服务器。

Web 服务器插件（例如负载均衡器插件）可使您在一个或多个 Web 服务器的后面部署应用服务器，这些 Web 服务器位于受到一层或多层防火墙保护的隔离区 (DMZ) 中。这些插件为前端 Web 服务器层提供了将接收自 Internet 的流入的 HTTP 和 HTTPS 通信定向到位于后端应用服务器层中的一个或多个应用服务器的方法。

有多种客户机应用程序可以访问部署到应用服务器的商业服务。Web 服务客户机和基于浏览器的客户机可以使用 HTTP 或 HTTPS 访问 Web 服务、服务器端的端点和 J2EE Web 应用程序。

图 1-1 详细地显示了一个应用服务器实例。应用服务器实例是群集功能、负载均衡功能和会话持久性功能中的生成块，第 18 页上的“关于企业版功能”中介绍了这些功能。

图 1-1 Sun Java System Application Server 实例



关于管理域

管理域允许您定义多个完全独立的应用服务器运行时配置，这些配置重复使用相同的安装映像。每个管理域均有一个管理服务器，该管理服务器控制一个或多个应用服务器实例。图 1-1 显示了一个管理域。

在下面的教程中，您将使用产品安装期间配置的一个管理域和群集中的多个服务器实例。

关于企业版功能

Sun Java System Application Server 企业版 7 2004Q2 提供了先进的群集和故障切换技术。这些功能可以帮助您运行可扩展的且具有高可用性的 J2EE 应用程序。

本指南向您展示了如何使用以下各部分中介绍的企业版功能：

- [群集](#)
- [会话持久性](#)
- [具有高可用性的数据库](#)

群集

一个群集就是一组应用服务器实例。使用时，这些实例是一个逻辑整体。群集中的每个应用服务器实例都拥有相同的配置，并被部署了相同的应用程序。有关应用服务器群集的详细介绍，请参见第 29 页上的第 2 章“群集方案”。

群集中的应用服务器实例可以存储在不同的计算机上，也可以存储在同一台计算机上。也就是说，可以将不同计算机上的应用服务器实例编组到一个逻辑群集中。在本指南中，缺省配置适用于同一台计算机上的两个实例。

有关群集的详细信息，请参见 *Sun Java System Application Server Administration Guide*。

在 Sun Java System Application Server 中使用群集可帮助您获得：

- [高可用性](#)
- [可扩展性](#)
- [负载均衡](#)

高可用性

Sun Java System Application Server 通过为群集中的应用服务器实例提供故障切换保护来提供高可用性。一个应用服务器实例出现故障时，另一个应用服务器实例将接管已指定到该故障服务器实例的会话。

会话信息存储在具有高可用性的数据库 (HADB) 中，该数据库与 Sun Java System Application Server 企业版捆绑在一起。有关 HADB 的详细信息，请参见第 26 页上的“具有高可用性的数据库”。

Sun Java System Application Server 为以下各项提供故障切换支持：

- [HTTP/S 会话](#)
- [HTTP 会话中的 EJB 引用](#)
- [通过 RMI/IIOP 进行 EJB 和 JNDI InitialContext 的远程引用](#)
- [JMS 连接](#)

HTTP/S 会话

服务于一个会话的某个应用服务器实例变得不可用时，负载均衡器插件会将 HTTP/S 连接及关联的会话信息故障转移到另一个服务器实例。启用高可用性后，会话信息存储在 HADB 中。

负载均衡器插件通过以下两种方法跟踪 HTTP/S 会话：

- Cookie
- 明确的 URL 重写

群集配置应创建于 loadbalancer.xml 文件中，该文件需要根据与负载均衡器插件一起安装的 loadbalancer.xml.example 文件来手动创建。默认情况下，此示例配置文件位于 Web 服务器的 config 目录中。

有关配置 HTTP/S 会话的高可用性的详细信息，请参见第 55 页上的第 4 章“[Cluster JSP 样例应用程序教程](#)”。

HTTP 会话中的 EJB 引用

本产品支持存储在 HTTP 会话中，用于实现 java.io.Serializable 界面的 EJB 引用和 J2EE 对象的故障切换。有关详细信息，请参见 *Sun Java System Application Server Administration Guide* 中的 "Configuring Session Persistence"。

通过 RMI/IIOP 进行 EJB 和 JNDI InitialContext 的远程引用

要启用 RMI/IIOP 应用程序的高可用性，需要用指定的 IIOP 端点配置一个 IIOP 群集。ORB 侦听程序也称作 IIOP 端点，它们将成为 IIOP 群集的一部分。IIOP 端点可以使用管理控制台或 CLI 进行配置。

沿 RMI/IIOP 路径接收到请求后，应用服务器将在群集中随机选取一个可用的 IIOP 端点作为主端点。然后，将群集中的其它 IIOP 端点指定为替代端点。

如果主端点变得不可用，与连接相关联的远程引用将被故障转移到替代端点之一。

IIOP 群集可以使用管理控制台或 CLI 进行配置，所做更改会被记录到应用服务器配置文件 `server.xml` 中。有关详细介绍，请参见第 31 页上的“[RMI/IIOP 群集方案](#)”。

注意 IIOP 和 HTTP 群集必须由相同的应用服务器实例组成。

JMS 连接

Sun Java System Message Queue Enterprise Edition 3.5 SP1 提供了 JMS 连接故障切换。

可扩展性

Sun Java System Application Server 允许将应用服务器实例添加到群集，这为 J2EE 应用程序提供了较高的可扩展性，进而提高了系统的性能。将应用服务器实例添加到群集时无需中断服务。

HTTP 和 RMI/IIOP 负载平衡系统会将请求分布到群集中正常的应用服务器实例上。

负载平衡

负载平衡的目的是平均分配多个 Sun Java System Application Server 实例之间的工作量。您可以为沿 HTTP/S 和 RMI/IIOP 路径的应用程序请求配置负载平衡。

本部分包括以下主题：

- [HTTP 负载平衡](#)
- [RMI/IIOP 负载平衡](#)

HTTP 负载均衡

Sun Java System Application Server 会将传入的 HTTP 和 HTTPS 请求分布到群集中已配置的各个应用服务器实例上。负载均衡由捆绑的负载均衡器插件来完成，这些插件安装在支持的 Web 服务器上。

当新的 HTTP 请求发送到负载均衡器插件时，基于运用一个简单的 round robin 机制，该请求被转发到一个应用服务器实例；具有相同上下文根路径的后续请求，基于运用 cookie 或明确的 URL 重写，将被分配到服务于第一条请求的应用服务器实例。

要配置 HTTP 负载均衡和故障切换，需要将应用服务器群集指定到负载均衡插件。对配置的更改必须在 `loadbalancer.xml` 文件中进行。有关负载均衡器配置文件的详细信息，请参见第 45 页上的“[创建 loadbalancer.xml 文件](#)”。

您可以使用随 Sun Java System Application Server 一起提供的负载均衡器插件，也可以使用各种第三方硬件和软件负载均衡器。本指南介绍了捆绑的 HTTP 负载均衡器插件。有关负载均衡的详细信息，请参见 *Sun Java System Application Server Administration Guide*。

RMI/IIOP 负载均衡

Sun Java System Application Server 企业版 7 基于 JNDI InitialContext，提供了 RMI/IIOP 路径上的远程 EJB 引用的负载均衡。对于每个新初始化的 JNDI InitialContext，会从已配置的 IIOP 群集中选择一个新的目标应用服务器实例。

要为基于 RMI/IIOP 的请求启用负载均衡，需要在 RMI/IIOP 客户机应用程序中进行一些细微的代码更改，以启用负载均衡。捆绑的 ORB 为负载均衡提供了必要的基础架构。

系统支持以下两种 RMI/IIOP 客户机的负载均衡。

- 在应用程序客户机容器 (ACC) 中运行，且访问部署在应用服务器实例上的 EJB 的 Java 应用程序。
- 不在 ACC 中运行，但访问部署在应用服务器实例上的 EJB 的 Java 应用程序。

启用 RMI/IIOP 客户机中的负载均衡的设置根据客户机的类型有所不同。有关示例配置，请参见第 31 页上的“[RMI/IIOP 群集方案](#)”。有关设置各种 RMI/IIOP 客户机的详细信息，请参见 *Sun Java System Application Server Developer's Guide to Clients*。

会话持久性

Sun Java System Application Server 实例或服务器失败时，会话持久性可以确保 HTTP/S 或 EJB 会话会由另一个服务器实例接管。Sun Java System Application Server 支持以下各项的持久性：

- HTTP 会话
- HTTP 会话中的 EJB 引用
- 状态会话 Bean (SFSB)

具有高可用性的数据库 (HADB) 与 Sun Java System Application Server 捆绑在一起，可用作持久性存储库。有关具有高可用性的数据库的详细信息，请参见[第 26 页上的“具有高可用性的数据库”](#)。

本部分包括以下主题：

- [关于会话持久性类型](#)
- [关于会话持久性配置](#)
- [关于单一登录会话信息](#)
- [关于 SFSB 检查点](#)

关于会话持久性类型

Sun Java System Application Sever 支持三种类型的持久性：ha、file 和 memory。

- ha（高可用性）类型的持久性将会话信息存储在 HADB 中。需要故障切换功能的生产环境支持 ha 类型的持久性。

可以使用管理界面来选择 HADB 持久性存储库。请执行以下步骤，将 HADB 设置为持久性存储库：

- a. 打开服务器实例下的“可用性服务”组件。
- b. 转至“可用性服务”页面。
- c. 选中“实例级别可用性”框。
- d. 单击“保存”按钮。
- e. 单击“持久性存储属性”下的“属性”。

- f. 在“名称”字段中，键入 store-pool-jndi-name。
- g. 在“值”字段中，键入 HADB JDBC 资源的 JNDI 名称。假定的缺省名称为 jdbc/hastore。
- h. 单击“保存”按钮。
- i. 转至服务器实例页面。
- j. 应用更改并重新启动服务器。

配置 HADB 后，server.xml 文件中的元素分层结构如下所示：

```
<server name="server1" ... >
    ...
    <availability-service availability-enabled="true">
        <persistence-store>
            <property name="store-pool-jndi-name" value="jdbc/hastore"/>
        </persistence-store>
    </availability-service>
    ...
</server>
```

有关如何配置 HADB 持久性存储库的信息，请参见管理指南。

- file 类型的持久性定期将会话信息存储到文件中，但如果实例变得不可用，可能会丢失数据。需要故障切换功能的生产环境不支持 file 类型的持久性，该类型仅有时用在测试应用程序的开发环境中。

如果将本地文件系统用于 SFSB 状态持久性，那么将由 server.xml 文件中的 session-store 属性来确定 SFSB 状态的存储位置。例如：

```
<server name="server1" ... session-store="/export/sfsbstore">
```

- 服务器实例正常关闭时，memory 类型的持久性会将会话信息存储到文件中。服务器实例异常关闭时，它并不存储会话信息。需要故障切换功能的生产环境不支持 memory 类型的持久性。

本指南的练习中仅涉及 ha 类型的会话持久性。有关配置和使用其它类型的会话持久性的详细信息，请参见 *Sun Java System Application Server Administration Guide*。

关于会话持久性配置

安装 Sun Java System Application Server 并运行 `clsetup` 命令后，会话持久性信息被配置为缺省值。但是，您可以更改缺省值以满足对于性能、可靠性和高可用性的特殊需要。以下两个主题中介绍了这些选项：

- [设置持久性范围](#)
- [设置持久性频率](#)

设置持久性范围

如果将 HADB 用作持久性存储库，必须设置持久性范围以配置应存储的会话状态的数目。例如，每次保存数据时，要么存储整个会话，要么仅存储已修改的会话。您可以配置持久性设置，以便仅存储会话中被修改的属性。

有以下三个选项供您选择：

modified-session 仅存储已修改的会话。

session 每次将会话状态保存到 HADB 时保存整个会话。

modified-attribute 仅保存会话中已修改的属性。

设置持久性频率

如果使用 HADB 存储会话状态，可以配置将会话状态存储到 HADB 数据库中的频率。例如，可以选择在每个 Web 请求之后存储会话，以提供更新的会话状态的高可用性和可靠性；也可以选择指定的时间间隔后存储会话，以提供更好的性能。

有以下两个选项供您选择：

web-method 如果选择 `web-method` 持久性频率，将在每个 Web 请求结束时存储会话。

time-based 如果选择 `time-based` 持久性频率，将根据在 `server.xml` 文件（实例级别的配置）或 `sun-web.xml` 文件（应用程序级别的配置）的 `manager-properties` 元素的 `reapIntervalSeconds` 特性中定义的时间间隔来存储会话状态。

注意 仅可为 `ha` 类型的持久性选择持久性频率。

有关会话持久性各种配置的详细信息，请参见 *Sun Java System Application Server Administration Guide* 中的 "Configuring Session Persistence"。

关于单一登录会话信息

在单个应用服务器实例中，如果一个 Web 应用程序验证了用户，则运行在同一个实例上的其它 Web 应用程序不会对用户进行重新验证。这称为单一登录。

为了使会话故障转移到群集中另一个实例后，该功能仍然可用，必须将单一登录信息保留在 HADB 中。持久性在启用应用服务器实例的高可用性时被启用，同时以下特性将被添加到 server.xml 中的 virtual-server 元素：

```
<property name="sso-enabled" value="true"/>
```

运行 clsetup 命令时，将自动配置高可用性。

关于 SFSB 检查点

在 SFSB 的生命周期中，其状态被保存到预定义点处的持久性存储库中。

SFSB 检查点有五个不同级别可供启用：

- 服务器实例
- EJB 容器
- 应用程序
- EJB 模块
- SFSB 自身

要启用给定级别的 SFSB 检查点，它必须已在所有更高级别中启用。例如，要启用应用程序级别的 SFSB 检查点，必须在服务器实例级别和 EJB 容器级别启用该 SFSB 检查点。

有关如何配置 SFSB 检查点的详细信息，请参见 *Sun Java System Application Server 7 Developer's Guide to Enterprise JavaBeans Technology*。

具有高可用性的数据库

具有高可用性的数据库 (HADB) 用于存储 HTTP 和 SFSB 会话信息。高可用性是指无论由于硬件还是软件失败引起意外故障，系统都保持可用状态。HADB 是 JDBC 兼容数据库，它基于 “Always-On” 技术，能够提供 99.999% 以上的数据可用性，为在高负载的企业级应用服务器环境中提交各种类型的会话状态持久性提供了一个理想的平台。

HADB 通过对数据进行分段和复制来提供数据的高可用性。将数据库中所有的表进行分区，以创建大小大致相同的子集（称之为段）。该分段过程基于散列功能，该功能在数据库节点之间对数据进行分段和平均分布。每个段存储两次，一次在数据库中，另一次在镜像节点中。这确保数据具有容错和快速恢复功能。此外，如果一个节点失败或关闭，备用的节点可以接管，直到第一个节点再次处于活动状态。

HADB 在一个单独但紧密集成的永久性存储层中存储和检索所有状态信息。有关安装和配置 HADB 的详细信息，请参见《*Sun Java System Application Server 安装指南*》中的 “HADB 安装的准备工作”。

配置和管理应用服务器的工具

Sun Java System Application Server 提供了以下这些配置和管理 Sun Java System Application Server 的工具。您可以使用这些工具来启动和停止服务器，也可以使用这些工具来执行多种其它功能。教程中包含了使用其中一些工具执行特定配置任务的详细步骤。

- **asadmin 实用程序。** asadmin 实用程序使用命令行界面，可以用来在单个应用服务器实例或管理域中执行管理任务。

有关 asadmin 的详细信息，请参见 *Sun Java System Application Server Administration Guide* 中的 "Appendix A, Using the Command Line Interface"。

- **cladmin 命令。** cladmin 命令在群集中的所有应用服务器实例上同步运行一些特定的 asadmin 命令。使用 cladmin 命令可以确保对群集中的所有实例进行相同地配置，所以只要可能就应使用 cladmin 命令，而不是 asadmin 命令。cladmin 命令并不支持所有的 asadmin 命令。

有关 cladmin 的详细信息，请参见 *Sun Java System Application Server Administration Guide* 中的 "Appendix F, Using the cladmin Script for Administration"。

- **管理界面。**管理界面是一个基于 Web 的用户界面，可以用来配置管理服务器和单个的应用服务器实例，还可以用来查看单个应用服务器实例的日志文件。

有关管理控制台的详细信息，请参见 *Sun Java System Application Server Administration Guide* 中的 "Part 1, Server Basics and Administering Global Settings"。

- **clsetup 命令。**clsetup 命令使您能够轻松地设置群集，包括 HADB 数据库的初始配置。有关 clsetup 命令的完整说明，请参见 《*Sun Java System Application Server 安装指南*》。此外，您应参考随 clsetup 一起提供的手册页。

注意

服务器配置的某些方面和相应的界面仍会有所变化。在以后发行的产品中，可能将不稳定的界面替换为更清晰、更稳定的界面。多数服务器配置和管理界面将保持不变，或以兼容的方式进行一些更改，但是某些方面可能会有很大的更改。以后发行的产品文档中将清楚地介绍实际进行了哪些不兼容的更改。

群集方案

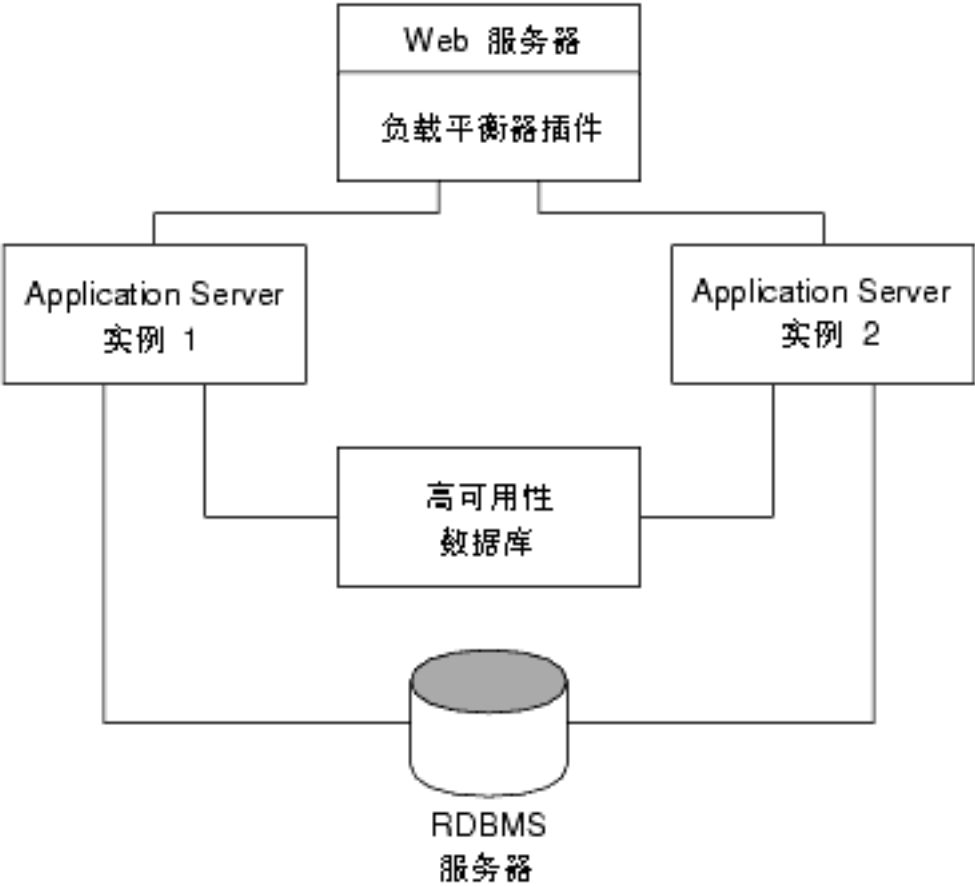
本章包括以下数节：

- [HTTP 群集方案](#)
- [RMI/IIOP 群集方案](#)

HTTP 群集方案

下面的图表显示了一个简单的群集方案，该方案由一个带有负载均衡器插件的 Web 服务器、两个 Sun Java System Application Server 实例（配置为使用 HADB 来存储 HTTP 会话数据）和一个用于存储应用程序数据的远程数据库管理系统 (RDBMS) 组成。请注意，您的实际操作部署可以有所不同。例如，您可以使用第三方的负载均衡器代替负载均衡器插件。

图 2-1 群集方案实例



下面的步骤介绍了处理 HTTP 请求的过程：

1. 连接的客户机将一个 HTTP 请求发送到某个由 Web 服务器负责处理的 URL，该 Web 服务器已配置为使用负载均衡器插件处理传入的 HTTP 请求。
2. 负载均衡器插件随后将请求转发到群集中的 Sun Java System Application Server 实例之一。插件使用粘性 round-robin（共享）负载均衡来确定目标实例。
3. 目标实例接收到转发自负载均衡器插件的请求后，开始一个 HTTP 会话，将 HTTP 会话数据和 J2EE 应用程序数据分别存储到 HADB 和 RDBMS。当客户机继续通过应用程序运行时，将更新 HTTP 会话数据并将其存储到 HADB；同时，RDBMS 中的应用程序数据也会更新。
4. 如果由于系统崩溃而导致实例失败，负载均衡器将检测实例是否已经停止对请求的响应。后续请求到来时，负载均衡器会将这些请求转发到群集中运行正常的实例。
5. 新的目标实例将检索来自 HADB 的故障转移 HTTP 会话信息，然后继续响应客户机的请求，使客户机能够完成 HTTP 会话，而不会丢失会话数据。

有关 Sun Java System Application Server 部署方案的详细信息，请参见 *Sun Java System Application Server System Deployment Guide*。

RMI/IIOP 群集方案

Sun Java System Application Server 通过在 RMI/IIOP 路径上使用负载均衡和故障切换机制，提供了具有高可用性的 J2EE 应用程序。

要配置具有高可用性的 J2EE 应用程序，需要在 IIOP 群集中配置 Application Server 实例，且该 IIOP 群集应具有用于处理 RMI-IIOP 请求的 IIOP 侦听程序。这些 IIOP 侦听程序也称作 IIOP 端点，它们将成为 IIOP 群集的一部分。

以下两部分介绍了在 RMI/IIOP 路径上启用 J2EE 应用程序的负载均衡（客户端配置）和高可用性（服务器端配置）所需进行的配置。

- [服务器端配置](#)
- [客户端配置](#)

服务器端配置

要启用 IIOP 请求的故障切换，应当配置 IIOP 端点，以组成 Sun Java System Application Server 中的 IIOP 群集。IIOP 端点可以使用管理控制台或命令行界面来定义。

首先定义群集中所有非 SSL 端点，RMI/IIOP 请求可以故障转移到该群集。availability-service 元素下的 iiop-cluster 特性定义了 IIOP 端点。然后确保设置了 availability-enabled="true"，以启用故障切换。

下面的示例显示了 server.xml 文件中的 IIOP 群集配置特性。

```
<availability-service availability-enabled="true">
  <iiop-cluster>
    <iiop-server-instance name=server1>
      <iiop-endpoint id=s1_ep1 host=trident port=3700 />
      <iiop-endpoint id=s1_ep2 host=trident port=3800 />
      <iiop-endpoint id=s1_ep3 host=trident port=3900 />
    </iiop-server-instance>
    <iiop-server-instance name=server2>
      <iiop-endpoint id=s2_ep1 host=jupiter port=4700 />
      <iiop-endpoint id=s2_ep2 host=jupiter port=4800 />
      <iiop-endpoint id=s2_ep3 host=jupiter port=4900 />
    </iiop-server-instance>
  </iiop-cluster>
</availability-service>
```


客户端配置

要启用 RMI/IIOP 请求的负载平衡，需要配置客户机应用程序。系统支持以下两种类型的 RMI/IIOP 客户机的负载平衡：

- 独立的客户机
- ACC 客户机

独立的客户机

要在独立的客户机中启用负载平衡功能，必须在 JNDI 环境特性或系统特性中定义以下特性。

- `java.naming.factory.initial`
将此特性设置为 `com.sun.appserv.naming.SlASCtxFactory`。
- `com.sun.appserv.iiop.endpoints`
指定 `server.xml` 中定义的 IIOP 端点列表。
- `com.sun.appserv.iiop.loadbalancingpolicy`
如果已指定端点，那么可使用此特性指定负载平衡的策略。用于定义此特性的值基于 JNDI `InitialContext`。

当 RMI/IIOP 客户机调用远程对象上的 `InitialContext` 查找时，将随机选择一个 **Application Server** 实例（该实例是 IIOP 群集的一部分）以创建 **Bean**。该 `InitialContext` 的所有后续操作都将被指定或固定到同一个 **Application Server** 实例。

要在 RMI/IIOP 客户机中实现负载平衡功能，请执行以下步骤：

1. 设置以下 JVM 特性以配置 ORB。
`com.sun.CORBA.connection.ORBSocketFactoryClass=com.sun.enterprise.iiop.EEIIOPSocketFactory`
`org.omg.PortableInterceptor.ORBInitializerClass=com.sun.enterprise.iiop.EEORBInitializer`
2. 将类路径设置为 `appserv-rt.jar` 和 `appserv-rt-ee.jar`。这些 jar 文件位于 `install_dir/lib` 目录中。

3. 在实例化 `InitialContext` 之前，使用 `S1ASCtxFactory` 类的以下特性：

```
Properties env = new Properties();

env.put("java.naming.factory.initial",
"com.sun.appserv.naming.S1ASCtxFactory");

env.put("com.sun.appserv.iiop.endpoints"."trident:3600,
exodus:3700");

env.put("com.sun.iiop.loadbalancingpolicy", "ic-based");

//create an initial naming context
Context initial = new InitialContext(env);
```

此客户机代码通过调用新的 `InitialContext(env)`（其中的 `env` 是 JNDI SPI 特性的列表），对 JNDI `InitialContext` 对象进行了实例化。

ACC 客户机

要在 ACC 客户机中启用负载平衡和故障切换功能，必须在 `sun-acc.xml` 中定义以下特性。

- `com.sun.appserv.iiop.endpoints`
- `com.sun.appserv.iiop.loadbalancingpolicy`

在 `sun-acc.xml` 文件中定义负载平衡特性，以启用具有高可用性的 ACC 客户机。这些特性在 `sun-acc.xml` 文件中被定义为特性元素。

例如：

```
<client-container>

    <target-server name="qasol-e1" address="qasol-e1" port="3700">

        <property name="com.sun.appserv.iiop.loadbalancingpolicy"
value="ic-based" />

        <property name="com.sun.appserv.iiop.endpoints"
value="qasol-e1:3700,jupiter:3800"/>

    </client-container>
```

企业功能配置教程

本章中的教程介绍了如何设置和使用 Sun Java System Application Server 环境中的功能。本章包括以下数节：

- [使用教程之前的准备工作](#)
- [教程步骤概述](#)
- [启动服务器](#)
- [检验服务器的启动](#)
- [创建 loadbalancer.xml 文件](#)
- [将群集添加到 loadbalancer.xml 文件](#)
- [配置负载平衡](#)
- [样例 loadbalancer.xml 文件](#)

使用教程之前的准备工作

本指南的内容包括设置群集环境、将应用程序部署到该环境中，以及用该应用程序测试负载平衡和 HTTP 会话持久性功能。本指南中有两个教程：[企业功能配置教程](#)介绍如何配置群集和负载平衡器；[Cluster JSP 样例应用程序教程](#)介绍如何部署应用程序及如何检验负载平衡和会话持久性已发挥作用。

本节包括以下主题：

- [安装需求](#)
- [使用教程之前需要执行的步骤](#)

安装需求

此处建议的安装配置是配合本指南使用的基本的单机配置，并不一定是恰好适合您的环境的配置。在您的环境中部署 **Sun Java System Application Server** 之前，请仔细考虑您自己的需要。有关建议配置的详细信息，请参见 *Sun Java System Application Server System Deployment Guide*。

为了使用本指南中包含的教程，请使用以下基本配置：

- 一台 **Sun Java System Web Server**
- 一个负载均衡器插件
- 一个 **Sun Java System Application Server** 安装，带有两个服务器实例，且已对这些实例进行配置以便使用具有高可用性的数据库。
- 一个具有高可用性的数据库

注意

本指南中的所有说明均假设您正在使用 **Sun Java System Web Server**。您也可以使用 **Apache Web 服务器**。有关使用 **Apache Web 服务器** 的详细信息，请参见 *Sun Java System Application Server Administration Guide*。

使用教程之前需要执行的步骤

开始使用教程之前，必须已完成以下步骤。

表 3-1 使用教程之前的准备步骤

步骤	相关说明的位置
1. 安装 Sun Java System Web Server	<i>Sun Java System Web Server Installation Guide</i>
2. 安装 Sun Java System Application Server，包括： <ul style="list-style-type: none">• 负载均衡器插件• HADB	《 <i>Sun Java System Application Server 安装指南</i> 》
3. 设置计算机中的以下内容： <ul style="list-style-type: none">• 共享的内存• 使用 rsh 或 ssh 的通信• 设置环境变量以使用 HADB	《 <i>Sun Java System Application Server 安装指南</i> 》
4. 运行 clsetup 以执行以下配置： <ul style="list-style-type: none">• 在域 domain1 中创建并配置两个应用服务器实例 server1 和 server2。• 创建 HADB。• 创建在 HADB 中存储会话信息所需的数据库表。• 创建实例中的连接池。• 创建所有实例中的 JDBC 资源。• 配置应用服务器实例中的会话持久性信息。• 启用应用服务器实例中的高可用性。	《 <i>Sun Java System Application Server 安装指南</i> 》

教程步骤概述

下表总结了本教程中配置系统、部署应用程序和检验企业功能所需的步骤。左侧列中介绍了步骤，右侧列中给出了执行该步骤所需说明的位置。

表 3-2 使用教程的步骤

步骤	所需说明的位置
1. 启动管理服务器和应用服务器实例。	第 39 页上的 “启动服务器”
2. 检验管理服务器和应用服务器实例正在运行。	第 41 页上的 “检验服务器的启动”
3. 创建 loadbalancer.xml 文件。	第 45 页上的 “创建 loadbalancer.xml 文件”
4. 在 loadbalancer.xml 中设置一个群集。	第 47 页上的 “将群集添加到 loadbalancer.xml 文件”
5. 在 loadbalancer.xml 文件中配置负载平衡器。	第 49 页上的 “配置负载平衡”
6. 利用样例 loadbalancer.xml 文件检验用户 loadbalancer.xml 文件。	第 53 页上的 “样例 loadbalancer.xml 文件”
7. 使用 cladmin 将样例应用程序部署到群集中的实例。	第 56 页上的 “将 Cluster JSP 样例应用程序部署到群集中”
8. 使用 cladmin 启动应用服务器实例。	第 61 页上的 “使用 cladmin 命令启动应用服务器实例”
9. 检验应用程序的部署是否成功。	第 61 页上的 “验证应用程序部署”
10. 在 loadbalancer.xml 中将样例应用程序添加到群集信息。	第 66 页上的 “将样例应用程序添加到群集”
11. 应用更改并重新启动 Web 服务器。	第 67 页上的 “应用配置更改并重新启动 Web 服务器”
12. 运行样例。	第 68 页上的 “运行应用程序”
13. 检验负载平衡。	第 73 页上的 “验证 HTTP 负载平衡”
14. 检验会话的持久性。	第 77 页上的 “验证 HTTP 会话持久性”
15. 停顿服务器实例。	第 78 页上的 “停顿服务器实例”

启动服务器

能够使用本指南中的教程之前，需要启动管理服务器和群集中所有的应用服务器实例。同时，Sun Java System Web Server 也须正在运行。

如果所有的服务器实例和管理服务器位于单个域中，那么启动管理服务器和应用服务器实例的最直接的方法是使用 `asadmin` 实用程序的 `start-domain` 命令。

本节包括以下主题：

- 设置 `PATH` 变量
- 运行 `asadmin start-domain`

设置 `PATH` 变量

为了方便起见，运行 `asadmin` 之前，可以设置 `PATH` 变量。

设置 `PATH` 变量可使您从命令行运行 `asadmin`、`cladmin` 和 `asant`（应用服务器的 `Ant` 实用程序），而无论您位于什么位置。请将下面的目录添加到 `PATH` 环境变量：

```
install_dir/bin
```

如果将 `Application Server` 的 `bin` 目录添加到登录配置文件中，则登录时该目录将被自动添加到环境的 `PATH` 设置中。

例如，将下面的内容添加到 `.cshrc` 文件：

```
set path=( /opt/SUNWAppserver7/bin)
```

保存文件后，对其执行 `source` 命令：

```
source .cshrc
```

您可以在命令提示符下，通过执行 `asadmin` 实用程序来测试所做的更改。请键入：

```
asadmin
```

应显示以下结果：

```
Use "exit" to exit and "help" for online help
```

```
asadmin>
```

请键入 `exit` 以退出 `asadmin` 界面。

如果该命令未找到：

- 请检查是否已正确地更新 `PATH` 设置。
- 如果更新了登录文件（例如 `.cshrc` 文件），请确保已对该文件执行了 `source` 命令。您可能需要启动一个新的终端窗口。

如果没有设置 `PATH` 环境变量，您仍然可以从这些命令和实用程序所在的目录中 (`install_dir/bin`) 运行它们。例如：

```
cd /opt/SUNWappserver7/bin
./asadmin
```

运行 `asadmin start-domain`

要启动管理服务器及其中的所有应用服务器实例，请运行 `asadmin start-domain` 命令。在命令提示符下，键入：

```
asadmin start-domain --domain domain1
```

`domain1` 是安装服务器和运行 `clsetup` 命令时配置的缺省域。

注意

在本指南介绍的配置中，所有的实例均位于同一个域中。如果服务器实例不是位于同一个域中，则需要为计划使用的每个域运行一次 `asadmin start-domain`。创建一个群集后，还可以使用 `cladmin start-instance` 来启动该群集中的所有实例。有关 `cladmin` 语法的信息，请参见第 58 页上的“[cladmin 命令的语法](#)”。

使用下面的命令停止管理服务器和最初配置的域的应用服务器实例：

```
asadmin stop-domain --domain domain1
```

其中，domain1 是在应用服务器安装过程中定义的管理域的名称。

有关 asadmin 命令的完整列表，请参见 asadmin 帮助。要访问帮助，请在命令提示符下键入：

```
asadmin
```

如果已设置 PATH 变量，提示应会更改为 asadmin 提示。请在提示下键入 help，以查看所有 asadmin 命令的列表。要获得关于某条特定 asadmin 命令的帮助，请在 asadmin 提示下键入该命令的名称，后接帮助选项。例如：

```
asadmin> start-domain --help
```

检验服务器的启动

启动管理服务器和应用服务器实例之后，请按照以下各部分中的说明检验它们是否已启动成功：

- [检验管理服务器的启动](#)
- [检验服务器实例的启动](#)
- [通过访问 HTTP 服务器来检验实例](#)

检验管理服务器的启动

可以通过访问 Sun Java System Application Server 的管理界面或通过查看管理服务器的事件日志来检验管理服务器的启动，如以下各部分所述：

- [访问管理界面](#)
- [查看管理服务器的事件日志](#)

访问管理界面

要检验管理服务器已正确启动，可以访问 Sun Java System Application Server 基于 Web 的管理界面。您可以使用此界面来管理 Sun Java System Application Server 的实例；但对于群集的实例，使用 `cladmin` 命令会更方便，引发错误的可能性也 smaller。

注意	有关与管理界面兼容的浏览器列表，请参见 <i>Sun Java System Application Server Platform Summary</i> 。
-----------	--

要访问管理界面，请执行以下步骤：

1. 打开一个浏览器窗口并指定管理服务器端口。

在安装过程中，管理服务器的缺省端口号被设置为 4848。如果此端口已被使用，或者您选择了其它端口号，请指定该端口号。

例如：

`http://test.sun.com:4848`

2. 使用产品安装过程中提供的管理用户名和口令登录到管理界面。

提示	忘记了用户名或口令？ 如果您忘记了安装过程中提供的管理服务器用户名，请尝试使用用户名 <code>admin</code> ，这是安装过程中在服务器配置对话框中指定的缺省用户名。
-----------	--

如果您忘记了管理员口令，请查看 `install_config_dir` 目录中的 `clpassword.conf` 文件，该文件是安装服务器时创建的。只有 `root` 用户才可读取此文件。

提示 端口不可访问？如果尝试连接管理服务器的管理界面时，连接被拒绝，可能是因为管理服务器没有运行。请检查启动过程和管理服务器日志文件的内容，以确定服务器没有运行的原因。有关查看日志文件的说明，请参见第 43 页上的“[查看管理服务器的事件日志](#)”。

验证成功之后，将显示管理界面的初始屏幕。单击左侧窗格中的某一项时，右侧窗格中将显示相应的页面。

查看管理服务器的事件日志

您可以在管理服务器的事件日志中查看服务器启动消息。要打开并查看事件日志文件，请执行以下步骤：

1. 浏览到管理服务器的服务器日志：

```
domain_config_dir/domain1/admin-server/logs/
```

2. 在编辑器中打开 `server.log`。

如果服务器已启动，您可以在日志文件的末尾找到 `successful server startup`：

```
[20/Feb/2003:00:06:00] INFO ( 4318):CORE3274:successful server startup
```

如果没有看到成功启动的消息，可能是因为您在管理服务器完成其启动过程之前，就打开了事件日志文件。请关闭日志文件然后重新打开，查看最新的事件消息。

您也可以使用 `tail -f` 命令来查看服务器日志中的消息。

```
tail -f server.log
```

`-f` 选项可使 `tail` 命令持续运行，以便在新的日志条目被写入文件时显示这些条目。

您还可以通过管理界面访问管理服务器和应用服务器实例的事件日志文件。单击左侧窗格中的服务器名称（管理服务器或应用服务器实例），然后单击右侧窗格中的“日志”选项卡。

有关详细信息，请参见 *Sun Java System Application Server Administration Guide*。

检验服务器实例的启动

运行 `clsetup` 之后，在 `domain1` 中应该有两个服务器实例：`server1` 和 `server2`。本部分讨论了检验这些实例存在且正在运行的两种方法：

- 使用 `asadmin` 检验实例
- 通过访问 HTTP 服务器来检验实例

使用 `asadmin` 检验实例

要检验实例，请使用 `asadmin list-instances` 命令：

1. 在命令提示符下键入 `asadmin`，以启动 `asadmin`：

```
asadmin
```

提示将变为 `asadmin>`

2. 在 `asadmin` 提示下使用 `list-instances` 命令显示所有当前实例。

```
list-instances
```

此命令将列出一些实例及其当前是否正在运行。您应该看到两个正在运行的实例 `server1` 和 `server2`。

通过访问 HTTP 服务器来检验实例

确定应用服务器实例是否已经启动的另一种方法是通过 Web 浏览器访问实例的 HTTP 服务器欢迎页。

使用浏览器访问以下位置：

`http://server_instance:server_instance_port_number`

其中，`server_instance_port_number` 是安装或创建服务器实例过程中指定的 HTTP 服务器端口号。

提示 如果忘记了服务器的 HTTP 服务器端口号，可以查看应用服务器实例的配置文件以确定 HTTP 服务器端口号：

1. 浏览到 *domain_config_dir*/domain1/*server_instance*/config/ 目录，然后在常用的编辑器中打开 *server.xml* 文件。

2. 查找 *http-listener* 元素，例如：

```
http-listener id="http-listener-1" address="0.0.0.0"
port="81"...
```

在本例中，端口 81 是正在使用的 HTTP 端口号。

如果应用服务器实例已启动并运行正常，您应该能在浏览器中看到默认的 HTTP 服务器欢迎页。

提示 **HTTP 服务器欢迎页：**HTTP 服务器欢迎页是一个名为 *index.html* 的 HTML 页面，位于应用服务器实例的默认文档目录中。应用服务器实例的 *server.xml* 配置文件包含了该实例默认文档根目录的设置。安装后，实例 *server1* 的文档根目录将被设置为 *domain_config_dir*/domain1/*server1*/*docroot*/。您可在该位置找到欢迎页。

创建 loadbalancer.xml 文件

安装组件、运行 *clsetup* 并启动管理服务器之后，您就可以配置该安装了。第一步是创建一个群集。

因为本指南的群集配置使用一个负载平衡器插件，所以必须在 Web 服务器的 *loadbalancer.xml* 配置文件中定义群集。此文件位于 Web 服务器的配置文件目录中。默认情况下，*loadbalancer.xml* 文件并不存在，您必须创建该文件。

要创建 loadbalancer.xml 文件，请执行以下步骤：

1. 在 Web 服务器的 config 目录中，找到名为 loadbalancer.xml.example 的示例 loadbalancer.xml 文件。

默认情况下，该目录为：

webserver_install_dir/servers/https-server_name/config

2. 将 loadbalancer.xml.example 的一个副本保存为 loadbalancer.xml。

您将编辑该副本以包含环境信息，包括创建群集和启用负载均衡功能的信息。

代码示例 3-1 中显示了一个示例 loadbalancer.xml 文件。

代码示例 3-1 loadbalancer.xml.example 文件

```
<!DOCTYPE loadbalancer PUBLIC "-//Sun Microsystems Inc.//DTD Sun ONE Application Server
7.0//EN" "sun-loadbalancer_1_1.dtd">

<loadbalancer>

  <cluster name="cluster1">

    <instance name="instance1" enabled="true" disable-timeout-in-minutes="60"
listeners="<REPLACE_WITH_LISTENER1> <REPLACE_WITH_LISTENER2>" />

    <web-module context-root="/abc" enabled="true" disable-timeout-in-minutes="60"
enabled="true" />

    <health-checker url="/" interval-in-seconds="10" timeout-in-seconds="30" />
  </cluster>

  <property name="reload-poll-interval-in-seconds" value="60"/>
  <property name="response-timeout-in-seconds" value="30"/>
  <property name="https-routing" value="true"/>
  <property name="require-monitor-data" value="false"/>
</loadbalancer>
```

请注意，示例文件指的是 sun-loadbalancer_1_1.dtd，它同样位于 Web 服务器的 config 目录中。

将群集添加到 loadbalancer.xml 文件

下一步，您需要将一个群集（本教程中称为 `cluster1`）添加到 `loadbalancer.xml` 文件。群集包含两个 Sun Java System Application Server 或服务器实例。

编辑新创建的 `loadbalancer.xml`，以在 HTTP 侦听程序中包含实例和 UTL：

1. 转到 Web 服务器的 `config` 目录并用文本编辑器打开 `loadbalancer.xml`。

请注意，样例文件已经包含群集 `cluster1`。

```
<cluster name="cluster1">
```

2. 将使用 `clsetup` 创建的两个应用服务器实例 `server1` 和 `server2` 添加到 `cluster1`。

为每个实例创建一个群集的实例子元素。需要启用实例 (`enabled=true`)，还需要添加 HTTP 侦听程序的 URL。对于默认的服务器实例 (`server1`) 和您添加的第二个服务器实例 (`server2`)，它们会各有一个 HTTP 侦听程序。在不同的环境中，可以拥有更多的侦听程序。

样例 `loadbalancer.xml` 文件包含以下这些可以编辑的行：

```
<instance name="instance1" enabled="true"
disable-timeout-in-minutes="60" listeners="<REPLACE_WITH_LISTENER1>
<REPLACE_WITH_LISTENER2>" />
```

3. 对于本教程，请用 “`server1`” 替换 “`instance1`”。
4. 用服务器实例的侦听程序的 URL 替换 `<REPLACE_WITH_LISTENER1>`，例如：

```
http://test.sun.com:81。
```

侦听程序的 URL 由服务器名和端口组成。如果侦听程序启用了安全性，则 URL 以 `HTTPS` 开头。如果侦听程序未启用安全性，则 URL 以 `HTTP` 开头。查找有关每个 Sun Java System Application Server 实例的 HTTP 侦听程序的信息的步骤：

- k. 在管理界面中，单击实例名称以展开树视图。
- l. 单击 “HTTP 服务器” 以展开该服务器。
- m. 在 “HTTP 服务器” 下，单击 “HTTP 侦听程序”。
- n. HTTP 侦听程序页将报告端口、服务器名以及是否启用了安全性。

5. 请将所做的更改保存到 loadbalancer.xml。此时文件会有与以下内容类似的一行：

```
<instance name="server1" enabled="true"
disable-timeout-in-minutes="60" listeners="http://test.sun.com:81"/>
```

6. 将 disable-timeout-in-minutes 的值更改为 5。

禁用超时（分）是实例的停顿时间，服务器实例的会话将在该时间段中终止。在此期间，服务器仍在运行，但只接受现有会话的请求，新会话生成的请求将不再发送到该服务器。在现实世界中，此数字可能会很高。示例中指定的时间是 60 分。但是，因为本教程需要您关闭服务器，所以要求您将超时值人为地设置得低一些。

```
<instance name="server1" enabled="true" disable-timeout-in-minutes="5"
listeners="http://test.sun.com:81"/>
```

7. 将第二个实例添加到群集元素。要完成此操作，您可以首先复制上一步中的行，然后将其粘贴到自身的下面，最后更改服务器名和侦听程序的 URL。例如：

```
<instance name="server2" enabled="true" disable-timeout-in-minutes="5"
listeners="http://test.sun.com:82"/>
```

8. 保存更改。

至此，loadbalancer.xml 文件应与下面的内容类似：

```
<!DOCTYPE loadbalancer PUBLIC "-//Sun Microsystems Inc.//DTD Sun ONE Application Server
7.0//EN" "sun-loadbalancer_1_1.dtd">

<loadbalancer>

  <cluster name="cluster1">

    <instance name="server1" enabled="true" disable-timeout-in-minutes="5"
listeners="http://test.sun.com:81"/>

    <instance name="server2" enabled="true" disable-timeout-in-minutes="5"
listeners="http://test.sun.com:82"/>

    <web-module context-root="/abc" enabled="true" disable-timeout-in-minutes="60"
enabled="true" />

    <health-checker url="/" interval-in-seconds="10" timeout-in-seconds="30" />

  </cluster>
```



```
<property name="reload-poll-interval-in-seconds" value="60"/>
<property name="response-timeout-in-seconds" value="30"/>
<property name="https-routing" value="true"/>
<property name="require-monitor-data" value="false"/>
</loadbalancer>
```

您将在以后的操作中添加与已部署的应用程序（以“web-module”开头的示例行）关联的信息。

注意

在 loadbalancer.xml 文件中启用应用服务器实例与启动应用服务器实例有所不同。即使服务器正在运行，且已将其添加到群集，负载均衡器也不会将请求路由到服务器实例，直到您将 instance 元素的 enabled 属性设置为 true，从而在 loadbalancer.xml 中启用了该服务器实例。

配置负载均衡

创建 loadbalancer.xml 文件并添加群集之后，请编辑 loadbalancer.xml 文件，以包含负载均衡器的设置。

本节包括以下主题：

- [配置运行状况检查器](#)
- [启用负载均衡器的监视功能](#)
- [负载均衡器的其它特性](#)

配置运行状况检查器

启用运行状况检查器后，负载均衡器将定期检查所有运行状况被标记为异常的 Sun Java System Application Server 实例，以查看其运行状况是否已变为正常。如果曾被标记为异常的应用服务器实例已变为正常，该实例将被添加到正常实例的列表，请求也将再次被路由到该实例。

运行状况检查器可以在 `loadbalancer.xml` 的 `health-checker` 元素中配置。您可以配置以下内容：运行状况检查器强制回应，以查看服务器是否正在运行的侦听程序的 URL；运行状况检查器强制回应服务器的频率；以及将服务器标为异常之前，运行状况检查器等待响应的的时间。

示例 `loadbalancer.xml` 文件包含以下这一行，您无需为了完成本指南中的任务而编辑该行。

```
<health-checker url="/" interval-in-seconds="10"
  timeout-in-seconds="30" />
```

- `health-checker url` 指定了要强制回应，以确定服务器实例是否正常（即，该实例是否响应）的 URL。示例显示了 `/` 的 URL。例如，如果侦听程序的 URL 是 `http://www.example.com:80`，那么 `/` 的运行状况检查器 URL 将强制回应 `http://www.example.com:80/`。在本指南中，请保留值 `/`。
- `interval-in-seconds` 属性指定了负载均衡器进行运行状况检查之间的时间间隔。您无需更改缺省值 10。
- `timeout-in-seconds` 属性指定了负载均衡器必须从强制响应的服务器实例接收响应，以便将该服务器实例视为正常的时间间隔。您无需更改缺省值 30。

例如，使用缺省值，运行状况检查器将执行以下步骤：

1. 运行状况检查器使用运行状况检查器的 URL 强制回应异常的侦听程序。
2. 对于每个侦听程序，运行状况检查器都将等待 30 秒 (`timeout-in-seconds`) 以获得响应。

如果服务器在 30 秒内响应，则侦听程序将被标记为正常。如果服务器没有响应，运行状况检查器将认为该服务器仍处于异常状态。

3. 在启动下一个运行状况检查周期之前，运行状况检查器将等待 10 秒 (`interval-in-seconds`)。

如果运行状况检查器启动时没有异常实例，运行状况检查器将跳过前两个步骤。

启用负载均衡器的监视功能

负载均衡器插件使用 Web 服务器的日志机制来生成日志消息。启用监视功能后，以下信息将被记录到 Web 服务器的日志文件中：

- 每个请求的开始/停止信息
- 请求从异常实例故障转移到正常实例时的故障切换信息
- 每个运行状况检查周期结束时记录的异常实例列表

警告

在负载均衡器插件上启用日志后，如果将 Web 服务器日志级别设置为 DEBUG 或设置为打印冗余消息，那么负载均衡器会将 HTTP 会话 ID 写入 Web 服务器日志文件中。所以，如果负载均衡器插件的宿主 Web 服务器位于 DMZ 中，那么建议不要在生产环境中使用 DEBUG 或类似的日志级别。

如果必须使用 DEBUG 日志级别，则应在 loadbalancer.xml 中将 require-monitor-data 特性设置为 false，从而关闭负载均衡器日志。

要监视负载均衡，需要进行以下操作：

- 在 loadbalancer.xml 中启用监视功能
- 配置 Web 服务器以使用冗余日志。

要在 loadbalancer.xml 中启用监视功能，请执行以下步骤：

1. 使用文本编辑器打开 loadbalancer.xml。
2. 找到以下行（示例 loadbalancer.xml 文件的一部分）：

```
<property name="require-monitor-data" value="false"/>
```
3. 将“false”更改为“true”以启用监视功能。
4. 保存更改并退出文件。

此外，如果要更改默认的 Web 服务器日志级别，请执行以下步骤：

打开 Sun Java System Web Server 的冗余日志：

1. 转到 Web 服务器的 config 目录。如果 Sun Java System Web Server 安装在缺省位置，则该路径是：

```
/web_server_install_dir/https-server_name/config
```

2. 打开 magnus.conf 以进行编辑。
3. 添加以下行：
LogVerbose on
4. 将更改保存到文件。
5. 重新启动 Web 服务器以应用这些更改。

负载均衡器的其它特性

示例 loadbalancer.xml 文件中包含了负载均衡器的一些其它特性：

```
<property name="reload-poll-interval-in-seconds" value="60"/>
```

```
<property name="response-timeout-in-seconds" value="30"/>
```

```
<property name="https-routing" value="true"/>
```

在本教程中，您无需更改这些缺省值。有关这些特性的详细信息，请参见以下各部分：

- [使用重新装入轮询时间间隔来进行动态重新配置](#)
- [响应超时](#)
- [HTTPS 路由选择](#)

使用重新装入轮询时间间隔来进行动态重新配置

完成初始配置后，负载均衡器插件将检测对其配置所做的更改并自动装入这些更改。负载均衡器插件检测这些更改的方式是查看 loadbalancer.xml 文件中的时间戳。如果时间戳已更改，负载均衡器将自动进行自我重新配置。重新装入轮询时间间隔指定了负载均衡器检查时间戳的频率。

注意

- 如果对 loadbalancer.xml 文件进行更改时使用的格式不是 sun-loadbalancer_1_0.dtd 文件中指定的正确格式，重新配置操作将会失败。失败通知将记录到 Web 服务器的错误日志文件中。

负载均衡器将继续使用已装入内存的旧配置。

- 如果尝试进行自我重新配置时，负载均衡器遇到了硬盘读取错误，它将使用内存中的当前配置。

遇到磁盘读取错误后，相关的警告消息将被记录到 Web 服务器的错误日志文件中。

Sun Java System Web Server 的错误日志文件位于：
web_server_install_dir/web_server_instance/logs/。

响应超时

响应超时指定了服务器等待实例响应请求的时间。超过指定的秒数后，如果实例仍未响应，将向浏览器发送一条错误消息。

HTTPS 路由选择

禁用 HTTPS 路由选择后，系统将忽略 loadbalancer.xml 文件中列出的 HTTPS 侦听程序，而仅对负载均衡使用 HTTP 侦听程序。启用 HTTPS 路由选择后，HTTPS 请求将仅故障转移到 HTTPS 端口。

样例 loadbalancer.xml 文件

以下的样例 loadbalancer.xml 文件包含一个带有两个实例的群集。请对照本示例，检查您的 loadbalancer.xml 文件的语法。

代码示例 3-2 The loadbalancer.xml 文件

```
<!DOCTYPE loadbalancer PUBLIC "-//Sun Microsystems Inc.//DTD Sun ONE Application Server
7.0//EN" "sun-loadbalancer_1_1.dtd">

<loadbalancer>
  <cluster name="cluster1">
    <instance name="server1" enabled="true" disable-timeout-in-minutes="5"
listeners="http://test.sun.com:81"/>
    <instance name="server2" enabled="true" disable-timeout-in-minutes="5"
listeners="http://test.sun.com:82"/>
    <web-module context-root="/abc" enabled="true" disable-timeout-in-minutes="60"
enabled="true" />
    <health-checker url="/" interval-in-seconds="10" timeout-in-seconds="30" />
  </cluster>
  <property name="reload-poll-interval-in-seconds" value="60"/>
  <property name="response-timeout-in-seconds" value="30"/>
  <property name="https-routing" value="true"/>
  <property name="require-monitor-data" value="true"/>
</loadbalancer>
```

现在，企业功能已经配置完毕，您可以部署和运行样例应用程序了。请继续阅读[第4章“Cluster JSP 样例应用程序教程”](#)。

Cluster JSP 样例应用程序教程

本章演示了使用负载均衡器插件进行的简单的 HTTP 故障切换。本章并未介绍 HTTP 负载均衡器插件的全部功能，有关负载均衡器插件的其它功能的信息，请参阅 *Sun Java System Application Server Administration Guide*。

本章包括以下几个部分：

- 使用 Cluster JSP 样例应用程序教程的准备工作
- 将 Cluster JSP 样例应用程序部署到群集中
- 使用 `cladmin` 命令启动应用服务器实例
- 验证应用程序部署
- 将样例应用程序添加到群集
- 应用配置更改并重新启动 Web 服务器
- 运行应用程序
- 验证 HTTP 负载均衡
- 验证 HTTP 会话持久性
- 停顿服务器实例

使用 Cluster JSP 样例应用程序教程的准备工作

在使用本教程之前，用户必须已执行了以下操作：

- 成功完成了第 3 章 “企业功能配置教程” 中的所有教程步骤。
- 创建了样例应用程序的副本（如有必要）。

如果您正与其他用户共享应用服务器安装，或者您的系统用户 ID 不具有对应用服务器安装区域的写入权限，您就需要创建样例的副本。

要复制样例，请将 *install_dir/samples* 目录复制到您的用户 ID 具有写入权限的位置。

例如：

```
cp -r install_dir/samples user_sample_directory
```

samples 目录包含子目录 *ee-samples*，该子目录包含了本指南中所使用的样例。

继续进行本指南时，如果您使用的是样例副本，则请将目录 *install_dir/samples/* 替换为您自己的样例应用程序副本的位置。

将 Cluster JSP 样例应用程序部署到群集中

Cluster JSP 样例应用程序演示了如何在群集中的应用服务器之间平均分摊对 JSP 的请求。其中的购物车样例演示了 HTTP 会话信息是如何保留下来的，即使应用服务器发生故障。

样例应用程序附带了 WAR 文件，该文件可以作为 Web 应用程序进行部署。用户无需编译或汇编样例应用程序。

注意

所有使用会话故障切换的应用程序都必须是可分布的。clusterjsp 样例应用程序是可分布的，如果您查看其 web.xml 文件就可以发现。

查看步骤：

1. 转到 Cluster JSP 样例的 src 目录：

```
cd /install_dir/samples/ee-samples/clustersjsp/src
```

2. 查看 web.xml 文件。您会看到以下代码：

```
<web-app>
    <display-name>clusterjsp</display-name>
    <distributable/>
```

web-app 元素具有指定的 distributable 子元素。无需更改应用程序的代码。

要将样例应用程序部署到群集中，必须先将它部署到群集中的每个实例中。

使用 cladmin 命令将应用程序同时部署到群集中的所有实例中。cladmin 命令将在群集中的所有实例上同时运行 asadmin 命令 cladmin 命令位于 *install_dir/bin* 目录中。

本部分包括以下主题：

- [cladmin 命令的输入文件](#)
- [cladmin 命令的语法](#)
- [运行 cladmin deploy 命令](#)
- [由 cladmin 命令支持的 asadmin 命令](#)
- [要求和限制](#)

cladmin 命令的输入文件

cladmin 命令使用两个输入文件：clinstance.conf 和 clpassword.conf。

- clinstance.conf: 此文件包含了关于群集组成部分中的应用服务器实例的信息。
- clpassword.conf: 此文件包含管理服务器的口令，并在标准安装过程中预先填入了正确的口令。

这些文件位于 *install_config_dir* 目录中。由于之前使用过这些文件来运行 clsetup，因此这些文件具有适合您的工作环境的正确值。

由于这些文件中包括了运行 cladmin deploy 命令需指定的多个值，因此可以用最少的选项来运行 deploy 命令。

cladmin 命令的语法

cladmin 命令的语法如下所示：

```
cladmin [--help] [--instancefile instance_file_location] [--passwordfile  
password_file_location] asadmin_command
```

其中：

- *instance_file_location* 是输入文件 clinstance.conf 所在的位置
- *password_file_location* 是输入文件 clpassword.conf 所在的位置
- *asadmin_command* 是希望在群集中的应用服务器实例上运行的 asadmin 命令。

如果输入文件位于默认位置 /etc/opt/SUNWappserver7 中，则可以省略 instancefile 和 passwordfile 选项，按以下所示运行此命令：

```
cladmin asadmin_command
```

运行 cladmin deploy 命令

要将应用程序部署到群集中的所有实例中，请键入：

```
cladmin deploy filepath
```

使用随样例提供的 WAR 文件，您可以将样例作为 Web 应用程序进行部署，其中 *filepath* 是 WAR 文件的路径。

例如，如果 Cluster JSP 应用程序位于默认位置，则键入：

```
cladmin deploy /opt/SUNWappserver7/samples/ee-samples/clusterjsp/clusterjsp.war
```

如果设置了 PATH 变量，您也可以转到 WAR 文件所在的目录，从该位置部署应用程序。例如：

```
cd install_dir/samples/ee-samples/clusterjsp
```

```
cladmin deploy clusterjsp.war
```

运行 cladmin deploy 命令时，您会在应用程序被部署到群集中的每个实例时看到相应的消息。

如果出现错误，请参考日志文件 `/var/tmp/cladmin.log` 以获得详细信息。

注意

如果您是以 root 用户身份操作，并且尚未设置 root 用户的 PATH 变量，那么您可能无法从任何目录中运行 cladmin 命令。请转到 *install_dir/bin* 目录，然后在命令提示符下键入 `./cladmin asadmin_command`。

由 cladmin 命令支持的 asadmin 命令

您可以使用 cladmin 命令，同时对群集中的每个应用服务器实例运行以下 asadmin 命令（configure-session-persistence 除外）：

表 4-1 由 cladmin 支持的 asadmin 命令

命令	用途
start-instance	启动应用服务器实例
stop-instance	停止应用服务器实例
deploy	在应用服务器实例中部署目录中的 EJB、WEB、连接器、appclient 或应用程序组件
undeploy	删除应用服务器实例中已部署的组件
create-jdbc-resource	为应用服务器实例创建 JDBC 资源
create-jdbc-connection-pool	为应用服务器实例创建 JDBC 连接池
configure-session-persistence	为应用服务器实例配置会话持久性。对于在同一台计算机上运行的 Application Server 的所有实例，只需运行一次此命令。
delete-jdbc-resource	删除应用服务器实例的 JDBC 资源
delete-jdbc-connection-pool	删除应用服务器实例的 JDBC 连接池

有关 cladmin 命令的详细信息，请参阅 *Sun Java System Application Server Administration Guide*。

要求和限制

cladmin 命令的运行有以下要求和限制：

- 群集组成部分中的所有实例必须具有相同的管理员口令。可以在 clinstance.conf 文件中的 user 属性名中指定用户名。
- 运行此命令之前，必须启动与群集组成部分中的所有应用服务器实例关联的管理服务器。
- 在输入文件中为群集中的所有实例指定的值是相同的。cladmin 命令不支持用不同的值来设置每个实例。例如，此命令无法为每个实例创建不同设置的 JDBC 连接池。

使用 cladmin 命令启动应用服务器实例

在部署完样例应用程序之后，请验证服务器实例处于运行状态。

1. 在命令提示符下，键入：

```
asadmin list-instances
```

此命令可以显示实例以及这些实例是否处于运行状态。

2. 如果实例尚未运行，请使用 cladmin 命令按以下所示启动这些实例：

```
cladmin start-instance
```

启动每个服务器实例时，您会看到相应的消息。

验证应用程序部署

部署了应用程序之后，您就可以在 Sun Java System Application Server 实例中运行该应用程序来测试部署结果。要运行样例应用程序，请执行以下步骤：

1. 通过浏览器访问以下 URL：

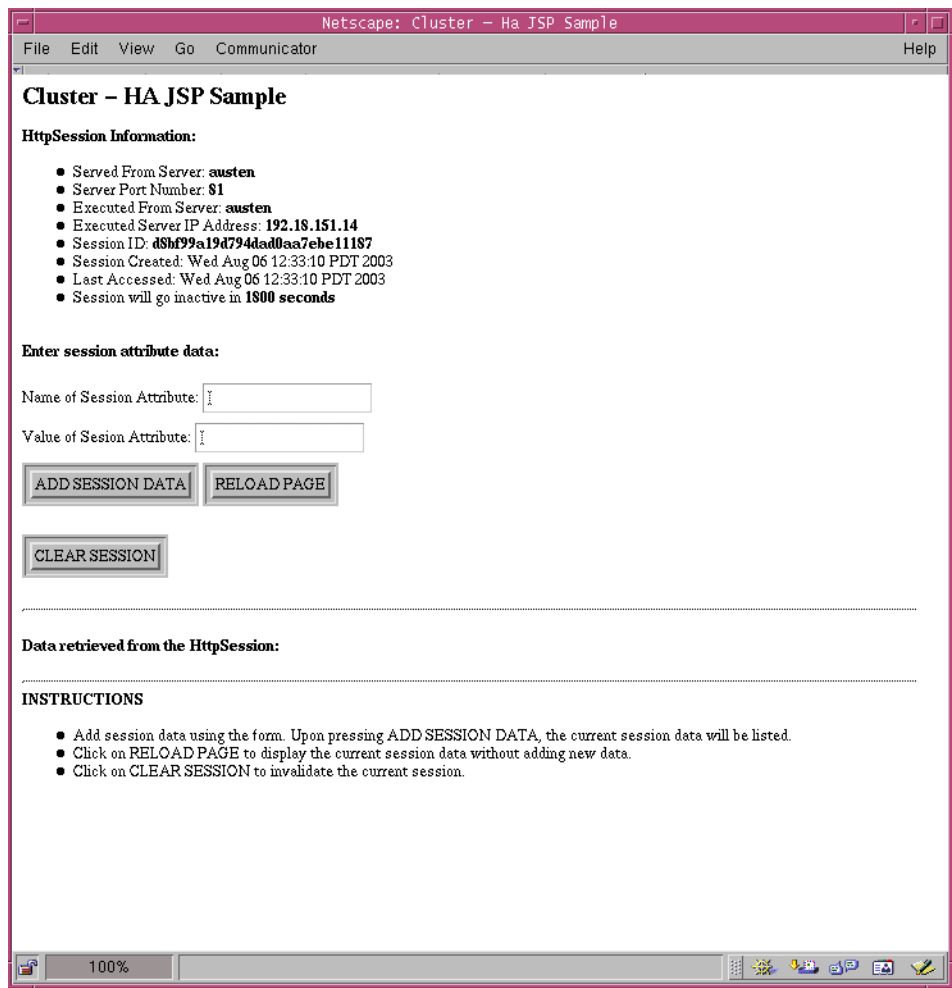
```
http://host:application_server_instance_port/clusterjsp
```

例如：

```
http://test.sun.com:81/clusterjsp
```

2. 您会看到此样例应用程序的页面。

图 4-1 Cluster JSP 样例应用程序页面



如果出现错误，请参见第 65 页上的“应用服务器实例部署的错误诊断”。

3. 在会话属性字段输入信息，然后单击“添加会话数据”。
应用程序将显示您所添加的数据。
4. 要验证下一个实例的部署，请在 URL 中键入该实例的端口和主机名：
`http://host:application_server_instance_port/clusterjsp`

例如：

`http://test.sun.com:82/clusterjsp`

这些 URL 可验证是否已将应用程序部署到应用服务器实例中。由于尚未将应用程序添加到 `loadbalancer.xml` 中的群集，因此还不能通过 Web 服务器验证该应用程序。

监视应用服务器中的样例

您可以通过查看应用服务器实例的事件日志文件来监视样例应用程序。每个应用服务器实例都有自己的事件日志。

本部分包括以下主题：

- [使用管理界面查看日志](#)
- [使用 `tail` 命令查看日志](#)
- [事件日志中由应用程序生成的消息](#)
- [访问日志中由应用程序生成的消息](#)

使用管理界面查看日志

要使用管理界面查看服务器实例的日志文件，请执行以下操作步骤：

1. 进入管理界面。
2. 在左侧的窗格中，单击希望检查其日志的实例。
3. 单击“日志”选项卡。
4. 单击“查看事件日志”
5. 要刷新日志视图，请单击“确定”。

如果要查看的日志条目超过 25 个，可直接在“要查看的错误数目？”区域中输入一个较大的数字，然后单击“确定”刷新日志。

要使用管理界面查看应用服务器实例的 HTTP 访问日志，请执行以下操作步骤：

1. 进入管理界面。
2. 在左侧的窗格中，单击希望检查其日志的实例。
3. 单击“日志”选项卡。
4. 单击“查看 HTTP 访问日志”。

服务器实例的 HTTP 访问日志中显示了服务器实例接收到的对样例应用程序的访问情况。

访问日志的名称为 `access`。默认情况下，访问日志文件和服务器事件日志文件位于同一目录中：

```
domain_config_dir/domain1/server1/logs/
```

使用 tail 命令查看日志

使用 `tail -f` 命令监视日志文件可以在记录这些文件后自动显示消息。您必须在管理界面中单击某个按钮来反映页面上的新信息。

要使用在中使用 `tail -f` 命令，请执行以下操作步骤：

1. 浏览到要监视的服务器实例的日志目录，例如：

```
cd domain_config/domain1/server1/logs/
```

2. 在服务器事件日志文件中运行 `tail` 命令：

```
tail -f server.log
```

`-f` 选项使 `tail` 命令处于运行状态，这样当有新的日志条目写入文件时，将显示该条目。

事件日志中由应用程序生成的消息

当应用程序向 `stdout` 和/或 `stderr` 写入信息时，默认情况下，该信息将作为 INFO 级的消息，用消息 ID `CORE3282` (`stdout`) 和 `CORE3283` (`stderr`) 记录到服务器实例的事件日志中。当运行 `Cluster JSP` 样例时，一些由应用程序生成的消息将被写入服务器事件日志。以下摘录的是某些消息的快照：

例如：

```
[13/Aug/2003:17:32:28] INFO ( 9657):CORE3282:stdout: 没有为该请求输入参数
[13/Aug/2003:17:32:34] INFO ( 9657):CORE3282:stdout: 添加到会话 :name1 = 1
[13/Aug/2003:17:32:45] INFO ( 9657):CORE3282:stdout: 添加到会话 :name2 = 2
[13/Aug/2003:17:32:52] INFO ( 9657):CORE3282:stdout: 添加到会话 :name3 = 3
```

这些消息显示出当向应用程序的用户界面输入数据时，值的变化。

访问日志中由应用程序生成的消息

运行 Cluster JSP 样例时，将在响应访问请求的应用服务器实例中记录访问日志消息。

```
192.18.151.14 - - [12/Aug/2003:15:34:52 -0700] "GET /clusterjsp/ HTTP/1.0"
302 1086
192.18.151.14 - - [12/Aug/2003:15:34:52 -0700] "GET /clusterjsp/HaJsp.jsp
HTTP/1.0" 200 1578
```

应用服务器实例部署的错误诊断

下表介绍了尝试运行此样例应用程序时最常见的问题。左侧列显示了出现的情况，中间列是可能导致问题的原因，右侧列是建议尝试的解决办法。

表 4-2 应用服务器实例部署的错误诊断

出现的情况	可能原因	解决办法
尝试访问首页时连接失败。	未启动应用服务器。 在 URL 中指定了错误的端口。	确保已经启动了应用服务器。 确定正确的 HTTP 服务器端口号。 有关详细信息，请参见第 39 页上的“启动服务器”。
访问主页面时出现 404 错误。	未部署应用程序。	请参见第 56 页上的“将 Cluster JSP 样例应用程序部署到群集中”。

解决问题时，监视应用服务器的日志文件是非常必要的。同时，查看 HTTP 访问日志文件，以验证 HTTP 请求是否按预期到达应用服务器也很重要。

将样例应用程序添加到群集

当您在群集中的两台服务器实例中部署了样例应用程序之后，您就必须将该应用程序添加到在 loadbalancer.xml 中创建的群集：

1. 转到 Web 服务器的 config 目录，并使用文本编辑器打开 loadbalancer.xml 文件。
2. loadbalancer.xml 样例文件包含以下行：

```
<web-module context-root="/abc" enabled="true"
disable-timeout-in-minutes="60" enabled="true" />
```

更改此行以反映 clusterjsp 样例应用程序：

```
<web-module context-root="clusterjsp" enabled="true"
disable-timeout-in-minutes="60"/>
```

此行将把 clusterjsp 应用程序添加到群集并启用该应用程序。

上下文根路径与在 Sun Java System Application Server 实例的 server.xml 文件中为应用程序设置的值是同一个值。

disable-timeout-in-minutes 属性被设置为 60。也就是说如果禁用应用程序，将会为服务器留出 60 分钟的应用程序停顿时间，以使其完成响应所有未处理的应用程序请求。之前我们设置了实例级的 disable-timeout-in-minutes 属性，它可以控制服务器实例的停顿时间。

3. 保存所做的更改。

此时 loadbalancer.xml 文件应类似于以下文件：

代码示例 4-1 样例应用程序经配置的 loadbalancer.xml 文件

```
<!DOCTYPE loadbalancer PUBLIC "-//Sun Microsystems Inc.//DTD Sun Java
System Application Server 7.0//EN" "sun-loadbalancer_1_0.dtd">
<loadbalancer>
    <cluster name="cluster1">
        <instance name="server1" enabled="true"
disable-timeout-in-minutes="5" listeners="http://test.sun.com:81"/>
        <instance name="server2" enabled="true"
disable-timeout-in-minutes="5" listeners="http://test.sun.com:82"/>
    </cluster>
</loadbalancer>
```

```

        <web-module context-root="clusterjsp" enabled="true"
disable-timeout-in-minutes="60"/>

        <health-checker url="/" interval-in-seconds="10"
timeout-in-seconds="30" />

    </cluster>

    <property name="reload-poll-interval-in-seconds" value="60"/>

    <property name="response-timeout-in-seconds" value="30"/>

    <property name="https-routing" value="true"/>

    <property name="require-monitor-data" value="true"/>

</loadbalancer>

```

应用配置更改并重新启动 Web 服务器

编辑完 `loadbalancer.xml` 文件之后，您需要启动 Web 服务器；如果 Web 服务器已经处于运行状态，则需要应用配置更改并重新启动 Web 服务器。

启动 Web 服务器的步骤：

1. 转到 `web_server_install_dir/https-instance_name`。

例如：

```
/usr/iplanet/servers/https-test.sun.com
```

2. 键入 `./start` 启动服务器

可能需要同时启动 Web 服务器的管理服务器，管理服务器位于 `web_server_install_dir/https-admin-serv`。

将更改应用到处于运行状态的 Web 服务器并重新启动服务器的步骤：

1. 在 Web 浏览器中键入相应的 URL 来访问 Web 服务器的服务器管理器。

```
http://web_server:web_server_admin_port
```

例如：

```
http://test.sun.com:8888
```

2. 从服务器管理器中选择所需的服务器，然后单击“管理”。
则将显示所选 Web 服务器实例的服务器管理器。
3. 同时显示一则消息，警告您已经手动更改了配置文件，因此需要应用这些更改。
4. 单击页面右上角的“应用”链接。
5. 单击“应用更改”以应用所做的更改并重新启动服务器。

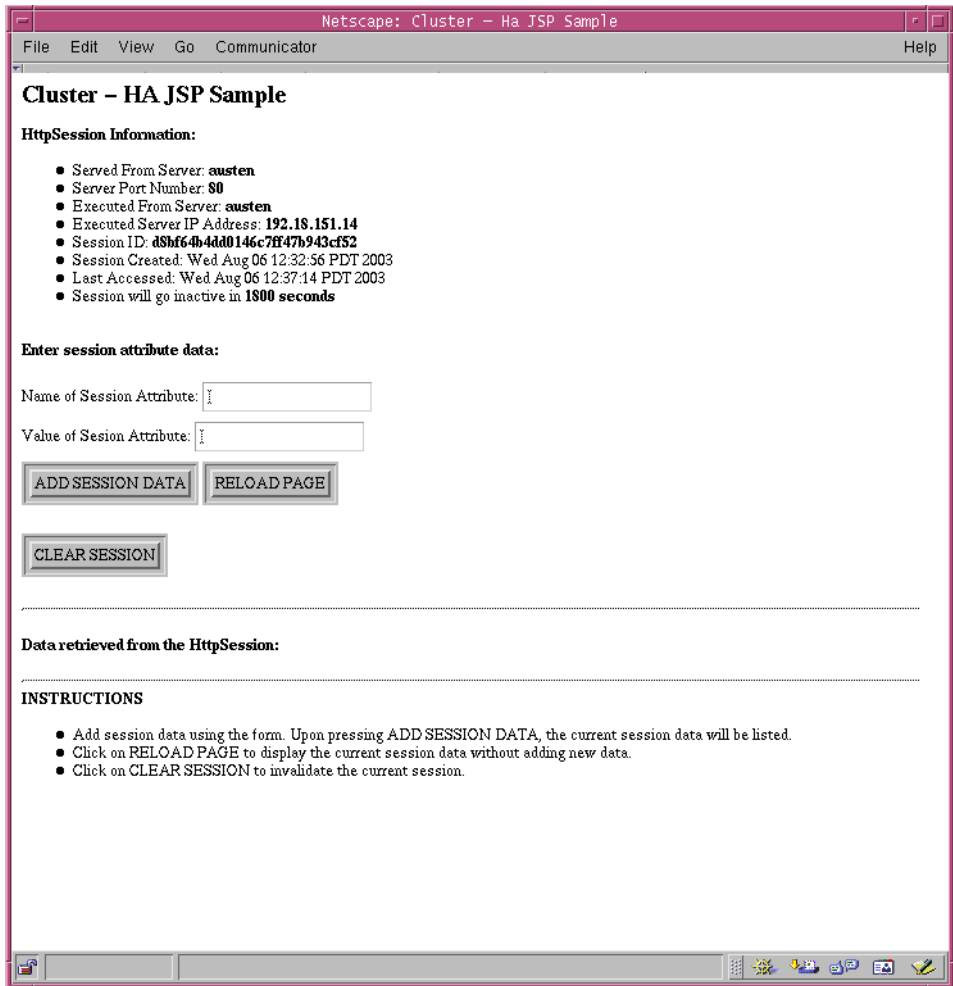
如果以后再更改 `loadbalancer.xml` 文件，就无需应用更改和重新启动服务器了。这时因为设置了重新装入轮询时间间隔，所以以后所做的所有更改都将被自动装入。有关详细信息，请参见第 52 页上的[“使用重新装入轮询时间间隔来进行动态重新配置”](#)。

运行应用程序

将应用程序部署到了所有实例中并更新了 `loadbalancer.xml` 文件之后，请在 Web 服务器上测试该应用程序以确保其正常工作。

1. 通过浏览器访问以下 URL：
`http://host:web_server_port/clusterjsp`
例如：
`http://test.sun.com:80/clusterjsp`
您会看到此样例应用程序的页面。

图 4-2 Cluster JSP 样例应用程序页面

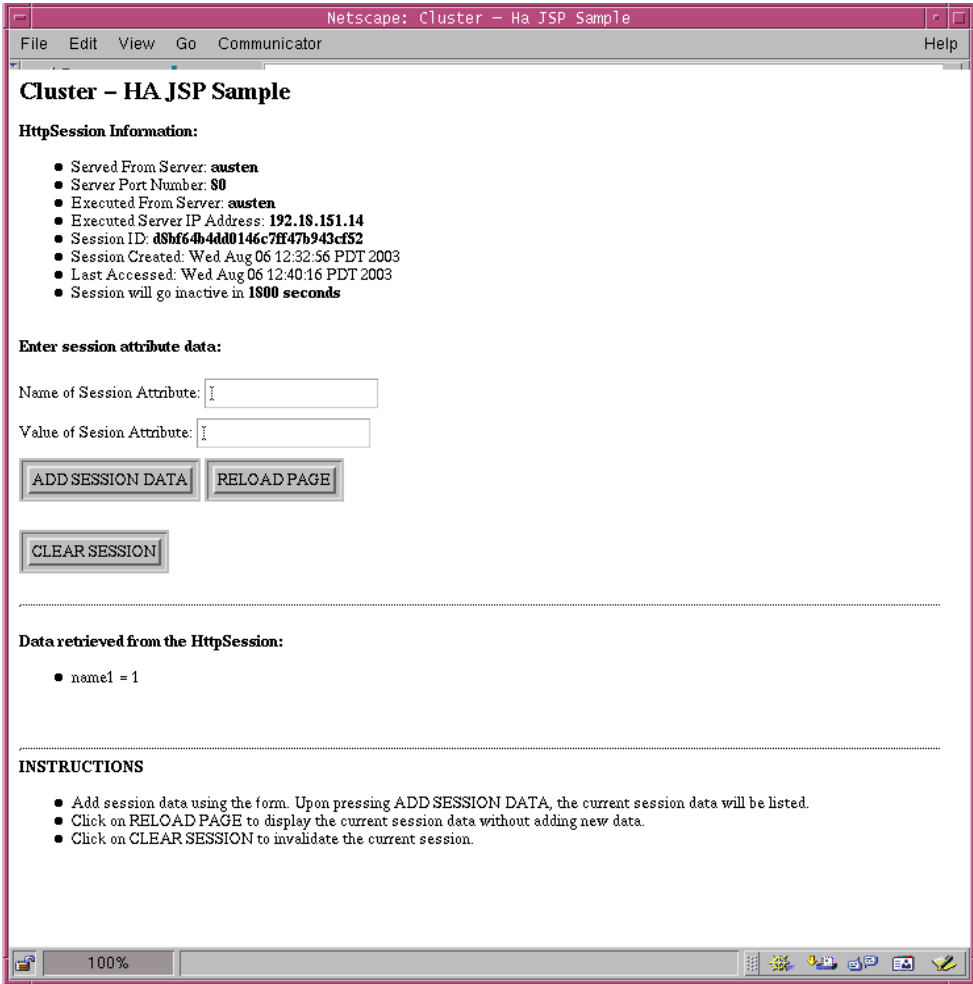


页面顶部列出了服务器信息，以及唯一的“会话 ID”。如果您直接在应用服务器实例上访问此应用程序，则页面顶部的端口信息是应用服务器的信息。如果您通过 Web 服务器访问群集的应用程序，则端口信息是 Web 服务器的信息。

2. 输入会话属性的名称和值，然后单击“添加会话数据”。

则页面上将显示输入的会话数据。

图 4-3 在 Cluster JSP 中显示的会话数据



请注意页面顶部的“会话 ID”。如果输入了会话属性并在会话超时前单击了“添加会话数据”，则页面顶部的“会话 ID”将与第一次访问应用程序时相同。

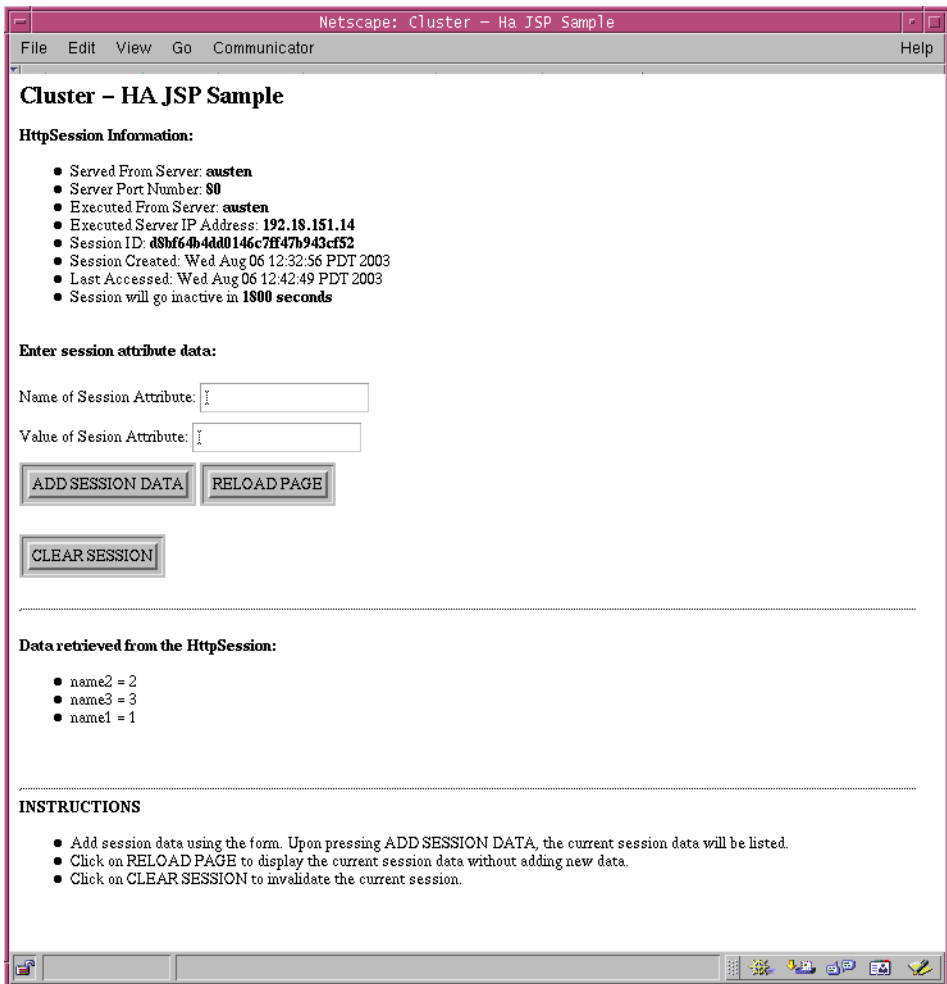
在上一个示例中，“会话 ID d8bf64b4dd0146c7ff47b943cf52”在两种情况下就是相同的。

但如果会话超时，“会话 ID”将会不同。

3. 输入新的会话属性名称和值。如果会话尚未超时，所有属性都会在“从 HTTP 会话中检索到的数据”区域中列出。

例如，如果在会话超时之前输入会话属性数据两次以上，则会显示类似于以下页面的页面。

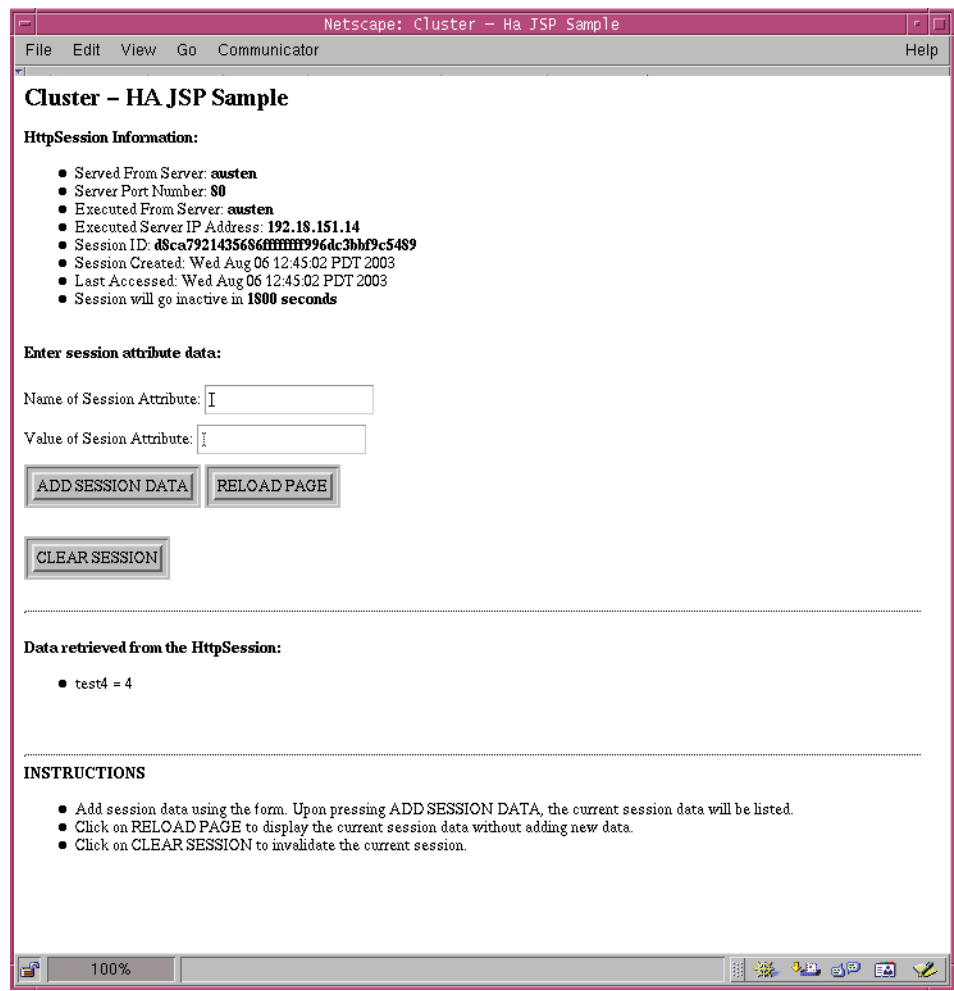
图 4-4 具有多个 HTTP 会话属性的 Cluster JSP 样例应用程序



4. 如果等到会话超时（1800 秒）以后再输入新的会话属性数据，则只显示最新输入的数据，并且页面顶部会显示新的会话 ID。

您也可以单击“清除会话”来模拟超时，这样会清除输入的会话数据，并为您提供一个新的会话 ID。

图 4-5 具有新的 HTTP 会话信息的 Clustr JSP



请注意页面顶部将显示一个新的会话 ID（本例中为“d8ca7921435686ffffff996dc3bbf9c5489”），并且“从 HTTP 会话中检索到的数据”部分仅显示最新输入的信息（本例中为“test4=4”）。

验证 HTTP 负载平衡

由于已经运行了样例应用程序，并且熟悉了会话信息的显示方式，因此您就可以验证负载平衡器是否处于工作状态。否则的话，您可以对负载平衡器进行错误诊断。

本部分包括以下主题：

- [验证负载平衡的步骤](#)
- [对负载平衡器插件进行错误诊断](#)

验证负载平衡的步骤

要验证负载平衡器是否处于工作状态，可执行以下操作步骤。

1. 打开两个单独的浏览器，然后键入
`http://web_server_name:web_server_port/clusterjsp`，以从 Web 服务器运行应用程序。例如：

```
http://test.sun.com:80/clusterjsp
```

由于使用负载平衡器时会话数据是粘性的，因此打开一个浏览器窗口访问应用程序时，该浏览器中所有的重新装入操作或其它操作均被认为处在同一会话中，并由一个 Sun Java System Application Server 实例来提供服务。

要验证应用程序的负载平衡是否处于工作状态，请再打开一个浏览器访问应用程序。由于会话数据保存在 `cookie` 中，因此您必须在另一台计算机上再打开一个浏览器，或者在同一台计算机上使用不同的浏览器软件。检查“会话 ID”以确保两个浏览器窗口有各自的会话数据。

2. 在两个单独的会话窗口中访问完了应用程序之后，请打开另一个浏览器窗口，查看 Web 服务器的错误日志。
 - a. 使用 Web 服务器文档中提供的 URL 访问 Sun Java System Web Server 的管理服务器，例如：
`http://test.sun.com:8888`
 - b. 选择所需的 Web 服务器实例，然后单击“管理”。
 - c. 在所选 Web 服务器实例的服务器管理器中，单击“日志”选项卡。
 - d. 在左侧的窗格中，单击“查看错误日志”。

- 3. 从错误日志中查找带有 RequestExit 的条目。
你可以直观地进行查找，也可以在“仅显示符合以下条件的项”字段中输入 RequestExit。
- 4. 带有 RequestExit 的条目应同时显示两个 Sun Java System Application Server 实例的 HTTP 侦听程序 ID，这样您就可以看到两个应用服务器实例都在响应请求。
- 5. 要验证 Sun Java System Application Server 实例已响应请求，请检查每个应用服务器实例的 HTTP 访问日志。
您应该会看到对 clusterjsp 应用程序发出的请求。有关检查访问日志的详细信息，请参见第 63 页上的“使用管理界面查看日志”。

注意 必须对 Web 服务器启用了冗余日志功能，才能查看 RequestExit 信息。相关说明请参见第 51 页上的“启用负载均衡器的监视功能”。

如果您在使用负载均衡器时遇到困难，请参阅下面的部分，该部分包含了有关对负载均衡器插件进行错误诊断的内容。

对负载均衡器插件进行错误诊断

使用下表可以帮助您对负载均衡进行错误诊断。左侧列说明出现的情况，中间列是可能导致问题的原因，右侧列是解决办法。

表 4-3 对负载均衡进行错误诊断

出现的情况	可能原因	解决办法
尝试访问应用程序时连接失败	未启动 Web 服务器	确保已启动了 Web 服务器。
	在 URL 中指定了错误的端口。	确定正确的 Web 服务器端口号。 请参阅 Web 服务器文档以获得详细信息。

表 4-3 对负载均衡进行错误诊断

出现的情况	可能原因	解决办法
访问应用程序时出现 404 错误	未部署应用程序。 loadbalancer.xml 文件中有错误	请参见第 56 页上的“将 Cluster JSP 样例应用程序部署到群集中”。 Web 服务器的错误日志中会记录 loadbalancer.xml 文件中的错误。有关详细信息，请参见第 75 页上的“查找 loadbalancer.xml 文件中的错误”。
访问应用程序时显示错误页面	群集中有一台或多台服务器未响应。	确保群集中的服务器实例处于运行状态。 检查错误日志以查看运行状况检查器是否已找到运行异常的实例。有关详细信息，请参见第 76 页上的“使用运行状况检查器”。

查找 loadbalancer.xml 文件中的错误

如果无法通过 Web 服务器 URL 访问到样例应用程序，可能是负载均衡器出了问题。这时，如果可以通过个别应用服务器访问到应用程序，就可以确定应用服务器实例没有问题。

由于负载均衡器插件会将信息记录到 Web 服务器日志文件中，因此您应该查看 Web 服务器错误日志文件以获得详细信息。

您可以通过服务器管理器查看 Sun Java System Web Server 的错误日志：

1. 使用 Web 服务器文档中提供的 URL 访问 Sun Java System Web Server 的管理服务器，例如：
`http://test.sun.com:8888`
2. 选择所需的 Web 服务器实例，然后单击“管理”。
3. 在所选 Web 服务器实例的服务器管理器中，单击“日志”选项卡。
4. 在左侧的窗格中，单击“查看错误日志”。
5. 默认视图显示 25 条信息；您可以根据需要增加显示信息的数目。

如果是负载均衡器出现了问题，您会看到包含“初始化负载均衡子系统失败”内容的信息。

如果是 loadbalancer.xml 文件有错误，您会看到解析器错误，它指向 loadbalancer.xml 文件中有错误的行。例如，您可能会看到与下面类似的信息：

```
[17/Jun/2003:09:57:44] catastrophe ( 5643):LBConfigParser.cpp@434: 报告
:lb.configurator:CNFG1000: 正在解析文件
:/usr/iplanet/servers/https-test.sun.com/config/loadbalancer.xml
```

第 7 行和第 3 列出错。错误消息为：内容模型不允许使用任何字符数据。

此信息指向 loadbalancer.xml 文件中有错误的行。

请在 loadbalancer.xml 文件中更正错误，应用更改并重新启动 Web 服务器，然后重试。

使用运行状况检查器

如果负载均衡器运行正常，并且已将 Web 服务器的 LogVerbose 设置为 on，则会在 Web 服务器的日志文件中看到运行状况检查器处于工作状态：

```
[09/Apr/2003:13:36:02] verbose ( 7576):HealthChecker.cpp@153: 报告
:lb.monitor:HLCK1006: 运行异常的实例 cluster1 1049920562631 无运行异常的实例
```

有关启用冗余日志功能的说明，请参见第 51 页上的“启用负载均衡器的监视功能”。

无论是否使用冗余日志，只要有一个实例出现异常，负载均衡器就会在访问失败时将该实例标记为运行异常。如果该实例仍处于异常状况，则运行状况检查器会再次将其标记为运行异常。

例如：

```
[17/Jun/2003:10:37:27] 警告 ( 5700): 报告 :lb.runtime:RNTM2024: 守护程序
http://test.sun.com:81 运行异常。
```

```
[17/Jun/2003:10:37:27] 警告 ( 5700):报告:lb.healthchecker:HLCK3003:检测到群集
cluster1 中的侦听程序 http://test.sun.com:81 仍处于异常状况
```

运行状况检查器将继续监视运行异常的实例，并在该实例运行正常后立即再将其标记为运行正常。

验证 HTTP 会话持久性

验证了负载均衡工作正常之后，就可以验证 HTTP 会话持久性了。在会话持久性作用期间，如果有一个 Sun Java System Application Server 实例由于某些原因而导致失败，则下一个实例将拾取该会话并处理其请求。

注意	会话持久性不在群集之间起作用，而仅在一个群集中的实例之间起作用。
-----------	----------------------------------

验证 HTTP 会话持久性在发挥作用的步骤：

1. 打开一个浏览器窗口。
2. 从 Web 服务器访问应用程序。例如：
`http://test.sun.com:80/clusterjsp`
3. 查看 Web 服务器错误日志，找到响应此请求的服务器（再次查找带有 RequestExit 的条目）。
4. 关闭提供此页面的 Sun Java System Application Server 实例。
您可以使用管理界面将其关闭：
 - a. 使用管理端口号来访问管理页面，例如：
`http://test.sun.com:4848`
 - b. 在左侧的窗格中，单击要停止的服务器实例。
 - c. 在右侧的窗格中单击“停止”。
5. 重新装入 Cluster JSP 样例应用程序页面。
会话 ID 和会话属性数据被保留。
6. 查看 Web 服务器错误日志。可以注意到另外一个 Application Server 的实例正在响应此请求。

服务器脱机后将保存会话信息。如果在应用程序传输数据时发生故障，可能会导致会话数据的丢失，这是唯一可能导致数据丢失的情况。在这种情况下，您可能会看到稍早一些的会话数据。

停顿服务器实例

在现实环境中，用户不会在未完成现有会话的情况下关闭服务器。相反地，用户会分阶段关闭服务器，这个过程称为停顿。

请按以下步骤停顿应用服务器实例：

1. 打开一个浏览器窗口，并通过 Web 服务器端口访问 Cluster JSP 应用程序。使此窗口保持打开状态。
2. 确定正在响应此请求的应用服务器实例。

打开另一个浏览器窗口，并在管理界面中访问 **server1** 应用服务器实例的 HTTP 访问日志。在日志中查找当您发出请求时 Cluster JSP 应用程序的访问记录。如果在 **server1** 的日志中看到访问记录，则说明 **server1** 应用服务器实例正在响应此请求。如果没有在访问日志中看到这样的条目，则请查看 **server2** 应用服务器实例的访问日志。使访问日志窗口保持打开状态。

3. 验证了正在响应此请求的应用服务器实例之后，请在 `loadbalancer.xml` 文件中禁用该服务器实例。

将相应服务器实例的 `enabled="true"` 改为 `enabled="false"`。例如：

```
<instance name="server1" enabled="false"
disable-timeout-in-minutes="5" listeners="http://test.sun.com:81"/>
```

4. 保存对 `loadbalancer.xml` 文件所做的更改。
5. 等到重新装入轮询时间间隔过后，将装入新的配置。

本教程中的重新装入轮询时间间隔为 60 秒，即只需等待一分钟。

6. 一旦重新装入轮询时间间隔已过，并装入了新的配置，负载均衡器插件就不再将不包含现有对话的请求发送到该服务器实例了。但是在停顿期间，带有现有会话的请求仍然会被发送到该服务器实例。

要查看此情况，请在打开的 Cluster JSP 窗口中输入名称和属性信息，然后单击“添加会话数据”

7. 转到一个浏览器窗口，该窗口显示了响应初始请求的服务器的 HTTP 访问日志。单击“确定”刷新信息。

请注意，尽管该服务器在 `loadbalancer.xml` 文件中被禁用，但它仍然在响应此请求。

8. 请等待五分钟，等到停顿时间结束后，在打开的 **Cluster JSP** 应用程序窗口中输入其它名称属性信息并单击“添加会话数据”。

服务器实例的禁用超时（以分钟为单位）控制着停顿时间。在本教程前面的部分，我们将禁用超时设置为五分钟。

9. 由于停顿时间已过，因此请求将被发送到其它应用服务器实例。

要查看此结果，请刷新服务器实例的 **HTTP** 访问日志。您会看到没有任何新请求。

10. 打开第三个浏览器窗口，查看其它服务器实例的 **HTTP** 访问日志。您会看到一个服务器实例正在响应此 **Cluster JSP** 请求。

然后，您就可以安全地关闭此应用服务器实例了。

您也可以停顿应用程序以使其安全脱机。有关应用服务器实例和应用程序的停顿的详细信息，请参阅 *Sun Java System Application Server Administration Guide*。

摘要和后续步骤

通过阅读本入门指南，您将了解 Sun Java System Application Server 的以下重要方面：

- 主要功能和体系结构
- 使用工具管理应用服务器
- 部署和使用 J2EE 应用程序
- 配置和使用负载平衡
- 配置和使用 HTTP 会话持久性
- 停顿服务器实例

查看下表中的后续步骤，确定您最感兴趣的内容。左侧的列包含感兴趣的主题，右侧的列包含完成该主题需要执行的后续步骤。

表 5-1 后续步骤

兴趣	后续步骤
深入了解 Sun Java System Application Server 样例应用程序中实现的企业版功能。	请在 <code>http://application_server_host_name:port/samples</code> 位置查看随 Application Server 安装的企业版的样例应用程序。索引页面提供了关于所有可用的企业版样例的信息。
了解设计和开发 J2EE 的最佳做法。	请参阅 Java BluePrints 并试用与应用服务器捆绑安装的 Java Pet Store 和 Smart Ticket 样例应用程序。
深入了解配置应用程序中的故障切换、会话持久性和高可用性会话。	请参阅 <i>Sun Java System Application Server Application Guidelines for Storing Session State</i> 并使用与应用服务器捆绑安装的 Duke's Bookstore 和 Session Storage 样例应用程序。

表 5-1 后续步骤

兴趣	后续步骤
了解管理 Sun Java System Application Server 7 的知识，包括群集、会话持久性、负载平衡和 HADB。	请参阅 <i>Sun Java System Application Server 7 Administrator's Guide</i> 。
通过使用教程深入了解 J2EE。	请参阅 J2EE Tutorial 和 Java Web 服务教程。
了解如何在 Sun Java System Application Server 7 上实现特定的 J2EE 功能。	请参阅应用服务器安装中所包括的样例应用程序。
了解使用 Sun Java System Application Server 7 进行开发的一般做法。	请参阅 <i>Sun Java System Application Server 7 Developer's Guide</i> 。

英文

- asadmin 实用程序 26, 60
 - configure-session-persistence 命令 60
 - create-jdbc-connection-pool 命令 60
 - create-jdbc-resource 命令 60
 - delete-jdbc-connection-pool 命令 60
 - delete-jdbc-resource 命令 60
 - deploy 命令 60
 - list-instances 命令 44, 61
 - PATH 变量设置 39
 - start-instance 命令 60
 - stop-instance 命令 60
- cladmin 命令 26, 57
 - deploy 命令 59
 - PATH 变量设置 39
 - start-instance 61
 - 日志文件 59
 - 位置 57
 - 限制 60
 - 要求 60
 - 语法 58
 - 支持的 asadmin 命令 60
- cladmin.log 文件 59
- clinstance.conf 文件 58
- clpassword.conf 文件 58
- clsetup 命令 27, 37
- Cluster JSP 样例应用程序 56
 - 运行 68
- configure-session-persistence 命令 60
- create-jdbc-connection-pool 命令 60
- create-jdbc-resource 命令 60
- delete-jdbc-connection-pool 命令 60
- delete-jdbc-resource 命令 60
- deploy 命令 60
- disable-timeout-in-minutes 48
- file 类型的会话持久性 23
- Forte for Java 12
- ha 类型的会话持久性 22
- HADB 26
- HTTP 访问日志 64
- HTTP 服务器
 - 端口 44
 - 欢迎页 45
 - 侦听程序 44
- HTTP 群集方案 29
- HTTP 侦听程序 47
 - 在 RequestExit 日志文件条目中 74
- http-listener 元素, server.xml 文件 45
- HTTPS 路由选择 53
- HTTPS 侦听程序 47, 53
- interval-in-seconds 50
- list-instances 命令 44, 61
- loadbalancer.xml 文件
 - 创建 45
 - 错误 75
 - 示例 46
 - 样例 53

A

- loadbalancer.xml.example 文件 46
- LogVerbose 52
- memory 类型的会话持久性 23
- modified-attribute 24
- modified-session 24
- PATH 环境变量, 设置 39
- RMI/IIOP 群集方案实例 31
- rpm 13
- session 24
- SFSB 检查点 25
- showrev 13
- start-domain 命令 39
- start-instance 命令 60, 61
- stderr 64
- stdout 64
- stop-instance 命令 60
- Sun Java Studio 12
- Sun 客户支持 13
- sun-loadbalancer.dtd 文件 46
- tail 命令 43, 64
- time-based 24
- timeout-in-seconds 50
- undeploy 命令 60
- Web 服务器
 - Apache 36
 - 错误日志 53
 - 配置日志 51
 - 应用更改 67
- Web 服务器错误日志 75
- web-method 24
 - 设置持久性频率 24

A

- 安装需求 36

B

- 部署
 - 错误诊断 65
 - 验证 61
 - 样例应用程序到群集 57

C

- 持久性频率
 - web-method 24
- 重新装入轮询时间间隔 52
- 错误日志, Web 服务器 53, 75
- 错误诊断
 - 负载均衡器 74
 - 样例应用程序 65

D

- 单一登录, 会话信息的可用性 25
- 单一登录会话信息 25
- 端口
 - HTTP 服务器 44
 - Web 服务器 68
 - 管理服务器 42
 - 无法访问 43

F

- 访问日志 65
- 服务器端 RMI/IIOP 故障切换的配置 32
- 服务器实例 16
 - 检验启动 44
 - 添加到群集 47
- 负载均衡器 20, 51
 - HTTPS 路由选择 53
 - 错误诊断 74

- 动态重新配置 [52](#)
- 特性 [52](#)
- 响应超时 [53](#)
- 样例 loadbalancer.xml [53](#)
- 运行状况检查器 [49](#), [76](#)

G

- 关于会话持久性类型 [22](#)
- 管理服务器 [16](#)
 - 端口 [42](#)
 - 检验启动 [41](#)
 - 事件日志 [43](#)
 - 用户名, 管理服务器
口令 [42](#)
- 管理界面 [27](#)
 - 查看日志 [63](#)
 - 访问 [42](#)
- 管理域 [18](#)

H

- 欢迎页, HTTP 服务器 [45](#)
- 会话持久性 [22](#)
 - file [23](#)
 - ha [22](#)
 - memory [23](#)
 - 配置 [24](#)
 - 验证 [77](#)
- 会话持久性类型 [22](#)
- 会话支持性配置 [24](#)

J

- 监视功能, 负载均衡器 [51](#)
- 监视样例应用程序 [64](#)
- 检验

- 服务器实例的启动 [44](#)
- 将更改应用到 Web 服务器 [67](#)
- 教程
 - Cluster JSP 样例 [56](#)
 - 步骤 [37](#)
 - 企业功能配置 [39](#)
 - 使用之前的准备工作 [35](#)
- 具有高可用性的会话持久性 [22](#)

K

- 可分布的应用程序 [57](#)
- 客户端 RMI/IIOP 负载均衡的配置 [33](#)
- 口令, 管理服务器 [42](#)

M

- 命令行实用程序 [26](#)

Q

- 启动应用服务器实例 [61](#)
- 启用会话持久性 [22](#)
- 群集 [18](#)
 - 部署应用程序 [57](#)
 - 创建 [47](#)
 - 方案实例 [29](#)
 - 添加应用程序 [66](#)

R

- 日志
 - HTTP 访问 [64](#)
 - Web 服务器错误 [53](#)
 - 查看 [63](#)

S

- 错误, Web 服务器 53
- 访问 65
- 启用 Web 服务器中的冗余日志 52
- 使用 tail 命令监视 43
- 使用 tail 命令进行监视 64
- 事件, 管理服务器 43
- 事件, 应用服务器实例 64

- 冗余日志 52

S

- 设置持久性范围 24
- 设置持久性频率 24
- 事件日志
 - 管理服务器 43
 - 应用程序消息 64
 - 应用服务器实例 43

T

- 停顿
 - 服务器实例 78
 - 应用程序 66

W

- 文档
 - 指南组织 10

X

- 响应超时 53

Y

- 验证
 - 应用程序部署 61
- 样例应用程序
 - Cluster JSP 56
 - WAR 文件 56
 - 部署到群集中 57
 - 错误诊断 65
 - 复制 56
 - 监视 64
 - 添加到群集 66
 - 验证部署 61
 - 运行 61, 68
- 应用程序, 样例
 - 监视 64
 - 添加到群集 66
 - 运行 61
- 应用程序, 样例
 - 错误诊断 65
 - 复制 56
 - 停顿 66
- 应用服务器
 - 检验启动 41
 - 启动 39
- 应用服务器实例 16
 - 检验启动 44
 - 添加到群集 47
 - 停顿 78
- 用户名, 管理服务器 42
- 域, 管理 18
- 运行状况检查器 49, 76
 - interval-in-seconds 50
 - timeout-in-seconds 50
 - URL 50