

<<Oracle WebLogic Serv>>

图书基本信息

书名：<<Oracle WebLogic Server开发权威指南>>

13位ISBN编号：9787302269823

10位ISBN编号：7302269823

出版时间：2011-11

出版时间：清华大学出版社

作者：[美]Robert Patrick,[美]Gregory Nyberg,[美]Philip Aston,[美]Josh Bregman,[美]Paul Done

页数：651

译者：贺新征,赵铭,任鸿

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<Oracle WebLogic Serv>>

### 内容概要

本书关注于如何通过最新发布的oracle weblogic server 11g进行java ee开发、部署和管理。

与其他介绍weblogic server和java

ee技术的书籍不同，本书的内容较为深刻，针对的是中高级读者。

而且，本书描述的是思路。

不只是简单地罗列出解决问题的各种可选方案，而后教会读者自己去做决策，还提供了在应用程序开发和管理过程中能够使用的具体建议和最佳实践。

本书构建并讲解的主要示例应用程序是一个真实复杂的应用程序，涵盖了很多java ee技术的许多特征，以及oracle weblogic server 11g特有的技术。

本书并不是java ee技术和weblogic server环境的入门读物。

这是一本高水平的书，对基本概念的描述很少，因此本书面向的读者是有经验的开发人员和weblogic server管理员，通过阅读本书他们可以将自己对这些技术的认识提升到更高水平。

## 作者简介

Robert Patrick是Oracle Fusion Middleware开发部门的副总裁，管理一个解决方案架构师团队(也称为A-Team)，负责EMEA(欧洲、中东和非洲)、APAC(亚太地区)和拉丁美洲地区的业务。Robert在设计和开发分布式系统领域拥有超过16年的丰富经验，他擅长设计利用多种中间件技术构建的大型高性能关键业务系统，并擅长对这类系统进行故障诊断。在加入Oracle之前，Robert在BEA Systems供职了7年半(最近的职位为Deputy CTO)。

## 书籍目录

## 第1章 在weblogic中构建web 应用程序

## 1.1 java servlet和jsp中的重要 概念

## 1.1.1 servlet的特征

## 1.1.2 jsp的特征

## 1.2 web应用程序最佳实践

## 1.2.1 保证正确的错误处理

## 1.2.2 使用jstl标记减少scriptlet 代码

## 1.2.3 对选定的行为使用自定义标记

## 1.2.4 对常见行为使用servlet过滤器

## 1.2.5 使用servlet和jsp页面创建 excel文件

## 1.2.6 查看产生的servlet代码

## 1.3 本章回顾

## 第2章 选择web应用程序体系结构

## 2.1 体系结构的关键概念

## 2.1.1 java ee应用程序的层次

## 2.1.2 mvc体系结构

## 2.1.3 通用java ee设计模式

## 2.2 表示层体系结构的选择

## 2.2.1 表示层需求

## 2.2.2 有关体系结构的其他考虑

## 2.3 可选的表示层体系结构

## 2.3.1 以jsp为核心的体系结构

## 2.3.2 以servlet为核心的体系结构：struts

## 2.3.3 以servlet为核心的体系结构：spring mvc

## 2.4 本章回顾

## 第3章 设计java ee应用程序示例

## 3.1 应用程序需求

## 3.2 业务领域模型

## 3.3 表示层需求

## 3.4 web应用程序体系结构

## 3.5 表示方法

## 3.5.1 自组装

## 3.5.2 母版页组装

## 3.5.3 使用tiles框架进行外部 视图组装

## 3.6 业务层接口

## 3.6.1 控制器调用业务服务

## 3.6.2 控制器为jsp填充表单

## 3.6.3 更新要求显式的服务调用

## 3.6.4 表示层组件之间的关系

## 3.7 本章回顾

## 第4章 构建web应用程序示例

## 4.1 应用程序组件概述

## 4.2 构建应用程序骨架

## 4.3 构建用户站点组件

## 4.3.1 预订信息组件

## &lt;&lt;Oracle WebLogic Serv&gt;&gt;

- 4.3.2 核心预订流程组件
- 4.3.3 提供服务组件
- 4.4 管理站点组件的结构
  - 4.4.1 身份验证和授权组件
  - 4.4.2 酒店维护组件
- 4.5 本章回顾
- 第5章 打包和部署weblogic web 应用程序
  - 5.1 打包web应用程序
    - 5.1.1 web应用程序目录结构
    - 5.1.2 web应用程序描述符文件
    - 5.1.3 预编译jsp组件
    - 5.1.4 创建一个分解式 web应用 程序
    - 5.1.5 创建一个web应用程序的 归档文件
  - 5.2 部署web应用程序
    - 5.2.1 自动部署
    - 5.2.2 weblogic部署工具和ant 任务
    - 5.2.3 weblogic控制台部署
    - 5.2.4 为bigrez.com创建必要的用户 和组
  - 5.3 本章回顾
- 第6章 在weblogic server中构建 企业javabean
  - 6.1 ejb技术概述
  - 6.2 ejb 3.0简介
  - 6.3 ejb组件类型
    - 6.3.1 无状态会话bean
    - 6.3.2 有状态会话bean
    - 6.3.3 消息驱动的bean
    - 6.3.4 拦截器
  - 6.4 java持久化api
    - 6.4.1 jpa的历史
    - 6.4.2 jpa概念
    - 6.4.3 一个jpa示例
    - 6.4.4 应用jpa
  - 6.5 weblogic server ejb容器
    - 6.5.1 ejb容器基础
    - 6.5.2 weblogic server中ejb的 生命周期
  - 6.6 通用的weblogic server ejb 特性
    - 6.6.1 ejb的部署和重新部署
    - 6.6.2 动态ejb编译
    - 6.6.3 ejb的远程业务接口和jndi
    - 6.6.4 ejb之间的引用
  - 6.7 会话bean的特性
    - 6.7.1 无状态会话ejb缓冲池
    - 6.7.2 有状态会话ejb的缓存管理
    - 6.7.3 有状态会话ejb的内存中 复制
    - 6.7.4 指向会话bean的句柄
    - 6.7.5 幂等方法
  - 6.8 消息驱动的bean的特性

## &lt;&lt;Oracle WebLogic Serv&gt;&gt;

## 6.9 openjpa和kodo特性

## 6.9.1 kodo部署描述符文件

## 6.9.2 获取组

## 6.9.3 预先获取

## 6.9.4 乐观锁版本策略

## 6.9.5 大结果集

## 6.9.6 二级缓存

## 6.9.7 控制刷新行为

## 6.9.8 逆向管理

## 6.9.9 混合继承策略

## 6.9.10 预处理语句缓存

## 6.10 使用部署描述符还是标注

## 6.10.1 部署计划

## 6.10.2 标注、描述符、计划和 依赖注入

## 6.11 本章回顾

## 第7章 构建ejb应用程序示例

## 7.1 业务层需求

## 7.1.1 业务逻辑需求

## 7.1.2 对象关系映射需求

## 7.1.3 数据访问需求

## 7.1.4 其他业务层需求

## 7.1.5 业务层需求的回顾

## 7.2 可选的业务层体系结构

## 7.2.1 slsb和会话外观模式的需求

## 7.2.2 使用jdbc的无状态会话 ejb

## 7.2.3 使用ejb 2.1 cmp实体bean 的无状态会话ejb

## 7.2.4 使用jpa的无状态会话ejb

## 7.3 bigrez.com的实现

## 7.3.1 数据库模式

## 7.3.2 域模型

## 7.3.3 服务

## 7.3.4 单元测试

## 7.3.5 添加乐观锁

## 7.4 使用toplink而不是kodo

## 7.4.1 希望使用toplink的理由

## 7.4.2 让bigrez.com使用toplink

## 7.5 本章回顾

## 第8章 打包和部署weblogic应用 程序

## 8.1 创建ejb归档文件

## 8.2 打包jpa持久化单元

## 8.3 企业应用程序

## 8.3.1 企业应用程序目录结构

## 8.3.2 企业应用程序的描述符文件

## 8.3.3 分解式部署

## 8.3.4 捆绑库

## 8.3.5 共享java ee库和可选包

## 8.3.6 其他类型的模块

## &lt;&lt;Oracle WebLogic Serv&gt;&gt;

- 8.3.7 自定义类加载器
- 8.4 打包bigrez.com
- 8.5 部署应用程序
  - 8.5.1 创建必需的服务
  - 8.5.2 自动部署和weblogic.deployer
  - 8.5.3 weblogic控制台部署
- 8.6 本章回顾
- 第9章 开发和部署web服务
  - 9.1 总结web服务标准
  - 9.2 用weblogic server创建web 服务
    - 9.2.1 web服务的容器框架
    - 9.2.2 开发weblogic server的web 服务
    - 9.2.3 开发web服务客户端
  - 9.3 高级技术
    - 9.3.1 使用jax-rpc
    - 9.3.2 理解style和use
    - 9.3.3 影响调用哪个操作
    - 9.3.4 创建更多动态的web服务
    - 9.3.5 使用web服务的处理程序
    - 9.3.6 使用soap附件
    - 9.3.7 实现有状态的web服务
    - 9.3.8 实现异步的web服务
    - 9.3.9 自定义java和xml之间的映射
  - 9.4 使用web服务安全
    - 9.4.1 定义安全策略
    - 9.4.2 传输级安全
    - 9.4.3 消息级安全
    - 9.4.4 web服务的安全配置
  - 9.5 在bigrez.com中添加web 服务
  - 9.6 本章回顾
- 第10章 使用weblogic jms
  - 10.1 jms的重要概念
    - 10.1.1 了解消息传递模型
    - 10.1.2 回顾jms api
  - 10.2 weblogic jms提供程序
    - 10.2.1 了解weblogic jms 服务器
    - 10.2.2 群集weblogic jms
    - 10.2.3 weblogic jms的客户端
    - 10.2.4 配置weblogic jms
  - 10.3 weblogic jms应用程序设计
    - 10.3.1 选择一种目标类型
    - 10.3.2 定位目标
    - 10.3.3 选择合适的消息类型
    - 10.3.4 压缩大消息
    - 10.3.5 选择一种消息确认策略
    - 10.3.6 设计消息选择器
    - 10.3.7 选择消息的过期策略

## &lt;&lt;Oracle WebLogic Serv&gt;&gt;

- 10.3.8 处理有害消息
- 10.3.9 处理消息顺序的问题
- 10.3.10 使用事务
- 10.3.11 使用多播会话
- 10.3.12 处理请求/响应样式的 消息交换
- 10.4 weblogic jms应用程序编程
  - 10.4.1 weblogic jms与ejb和 servlet的组合使用
  - 10.4.2 在服务器中使用异步消息
- 10.5 外部jms提供程序
  - 10.5.1 了解消息传递桥
  - 10.5.2 了解存储转发服务
  - 10.5.3 使用消息驱动的bean
  - 10.5.4 把外部jms对象映射到 weblogic jndi上
  - 10.5.5 集成oracle高级队列
  - 10.5.6 选择一种集成策略
- 10.6 本章回顾
- 第11章 使用weblogic安全性
  - 11.1 weblogic安全概述
  - 11.2 weblogic安全框架
    - 11.2.1 嵌入式ldap服务器
    - 11.2.2 安全领域和提供程序
  - 11.3 使用外部安全存储
    - 11.3.1 管理外部 ldap 身份 验证
    - 11.3.2 管理rdbms身份验证
  - 11.4 设置ssl/tls
    - 11.4.1 ssl和x.509证书概述
    - 11.4.2 获取x.509证书
    - 11.4.3 配置单向ssl
    - 11.4.4 配置双向ssl
    - 11.4.5 调试ssl的问题
  - 11.5 编写安全敏感的java客户端
    - 11.5.1 编写使用jaas的java 客户端
    - 11.5.2 编写使用ssl的java 客户端
  - 11.6 管理应用程序的安全
    - 11.6.1 应用程序安全模型
    - 11.6.2 配置java ee应用程序的 安全
    - 11.6.3 配置weblogic server应用 程序的安全
    - 11.6.4 启动weblogic server
  - 11.7 单点登录
    - 11.7.1 saml语言
    - 11.7.2 配置跨域安全和单点登录
    - 11.7.3 自定义身份验证提供程序
  - 11.8 本章回顾
- 第12章 在weblogic server中管理 和部署应用程序
  - 12.1 weblogic体系结构的重要 概念
    - 12.1.1 域体系结构
    - 12.1.2 weblogic server体系结构



## &lt;&lt;Oracle WebLogic Serv&gt;&gt;

- 12.1.3 weblogic server群集体系 结构
- 12.1.4 管理服务器
- 12.1.5 节点管理器
- 12.2 weblogic server管理的重要 概念
  - 12.2.1 服务器状态
  - 12.2.2 服务器自我健康监控
  - 12.2.3 网络信道
- 12.3 配置weblogic server的域
  - 12.3.1 建立一个新域
  - 12.3.2 配置服务器
  - 12.3.3 配置群集
  - 12.3.4 配置网络信道
  - 12.3.5 建立节点管理器
  - 12.3.6 操作系统的配置
  - 12.3.7 java虚拟机的配置
  - 12.3.8 web服务器插件的配置
  - 12.3.9 管理端口和信道的配置
  - 12.3.10 为weblogic server配置 应用程序
- 12.4 监视 weblogic server 应用 程序
  - 12.4.1 使用weblogic脚本工具
  - 12.4.2 使用弃用的命令行管理 工具
  - 12.4.3 通过weblogic控制台进行 监视
  - 12.4.4 通过jmx实现编程监视
  - 12.4.5 通过snmp监视
- 12.5 管理 weblogic server 应用 程序
  - 12.5.1 诊断应用程序的问题
  - 12.5.2 版本化应用程序
  - 12.5.3 处理故障情况
- 12.6 本章回顾
- 第13章 优化weblogic server的性能
  - 13.1 系统性能概述
    - 13.1.1 核心原则
    - 13.1.2 基于weblogic server的 应用程序的调优
  - 13.2 性能最佳实践
    - 13.2.1 可提高性能的设计方法
    - 13.2.2 理解 web 容器的最佳 实践
    - 13.2.3 理解ejb容器的最佳实践
    - 13.2.4 应用数据库访问的最佳 实践
  - 13.3 诊断性能问题
    - 13.3.1 准备好故障诊断
    - 13.3.2 识别瓶颈并消除瓶颈
    - 13.3.3 解决问题
    - 13.3.4 常见应用服务器性能问题
    - 13.3.5 java栈跟踪
  - 13.4 本章回顾
- 第14章 开发环境的最佳实践
  - 14.1 定义需要的硬件和软件

## &lt;&lt;Oracle WebLogic Serv&gt;&gt;

- 14.2 安装weblogic server软件
- 14.3 开发项目结构
- 14.4 优化开发周期
  - 14.4.1 分解目录开发
  - 14.4.2 fastswap
- 14.5 建立构建过程
  - 14.5.1 持续集成
  - 14.5.2 代码检查和报告工具
- 14.6 集成开发环境
  - 14.6.1 先决条件
  - 14.6.2 为bigrez.com配置eclipse
  - 14.6.3 为bigrez.com配置 jdeveloper
  - 14.6.4 通过ide进行调试
- 14.7 创建一个单元测试基础设施
  - 14.7.1 单元测试的重要性
  - 14.7.2 junit测试框架
  - 14.7.3 容器外测试
  - 14.7.4 测试web界面
  - 14.7.5 web服务
  - 14.7.6 性能和并发性测试
- 14.8 本章回顾
- 第15章 生产环境的最佳实践
  - 15.1 部署策略
    - 15.1.1 评价部署策略
    - 15.1.2 服务器部署策略
    - 15.1.3 单站点部署策略
    - 15.1.4 多站点部署策略
    - 15.1.5 设计多站点weblogic 群集
    - 15.1.6 实现跨多站点的群集
    - 15.1.7 在每一个站点实现一个 群集
  - 15.2 全局和本地流量管理
    - 15.2.1 使用负载均衡器
    - 15.2.2 配合weblogic server使用 本地负载均衡器
    - 15.2.3 配合weblogic server使用 全局负载均衡器
  - 15.3 生产环境的安全策略
    - 15.3.1 理解应用程序的数据流
    - 15.3.2 理解防火墙布局
    - 15.3.3 使用连接过滤器
    - 15.3.4 锁定web应用程序
    - 15.3.5 检查其他安全因素
    - 15.3.6 使用ssl硬件加速
  - 15.4 本章回顾

## 章节摘录

版权页：插图：发生这个问题的原因可能是出现了瓶颈，或者存在低效的上游组件或下游组件，还有可能是应用服务器本身的问题。

执行下面的流程来修复这个问题：（1）通过weblogic.Admin命令行管理工具请求一个GETSFATB和一系列PING操作，以确定应用服务器本身功能正常。

第12章讲解了这个工具的使用以及一些可用的命令行选项和参数。

因为GETSTAIIE和PING操作会经过应用服务器的正常执行队列，因此正常的返回时间表明服务器内部一切都正常。

不正常的返回时间表明存在潜在的问题需要进行分析。

（3）过长的执行队列可能表明系统缺少执行线程了。

如果所有的执行线程都在活动，而且CPU的使用率低，那么添加执行线程应该可以提升吞吐量，因此，检查工作管理器的最大线程数约束，如果需要的话，提高该约束。

（4）如果队列开始饥饿，但是添加执行线程并没有提升性能，那么可能出现了资源争用的情况。

由于CPU使用率低，线程可能耗费大量时间等待某些资源，通常情况下都是数据库连接。

在控制台中通过.IDBC监测机制检查高级别的等待者或长等待时间。

也许只需要在JDBC连接缓冲池中添加连接就能解决问题了。

（5）如果问题不在数据库连接，你应该定时获得JVM的线程转储，确定线程是否在不停地等待某个资源。

获得一组4份线程转储，每个线程转储相隔5~10秒，然后比较它们，判断是否有独立的线程在等待同一个资源足够长的时间，以至于出现在多个线程转储中。

有问题的线程可能在等待一个被另一个线程占用的资源，也有可能是在等待更新数据库中的同一个表。

JRockitLatencyAnalyzer可以很轻松地识别所有资源争用的问题，而不需要借用线程转储或其他类型的监测机制。

一旦找到了资源争用的问题，就可以采取正确的方法来解决这个问题。

（6）如果瓶颈不在应用服务器，那么问题根源可能在服务器上游，例如网络或Web服务器。

通过准备好的系统监测工具检查应用服务器上游的所有可能瓶颈，并且对这些组件进行故障诊断。

## <<Oracle WebLogic Serv>>

### 编辑推荐

《Oracle WebLogic Server开发权威指南》：介绍了选择Java EEI应用程序体系结构的技巧全面讲解了多种设计方案、体系结构、构建技术、部署选项和管理技术提供了一个真实的示例，涵盖了JSP、Spring MVC、EJB 3.0、JPA和JAN.WS等关键技术详细分析了在开发和部署示例应用程序时所做决策的方方面面展示了开发环境和生产环境的配置、管理和调优的最佳实践，探讨了WebLogic Server JWS和WebLogic安全性。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>