

<<大型门户网站是这样炼成的！>>

图书基本信息

书名：<<大型门户网站是这样炼成的！>>

13位ISBN编号：9787121102660

10位ISBN编号：7121102668

出版时间：2010-3

出版时间：电子工业出版社

作者：刘斌,飞思科技产品研发中心

页数：907

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>

前言

随着互联网技术的迅猛发展与Java语言的不断完善与优化，越来越多的Web程序员义无反顾地踏上了Java Web应用开发之路。

最原始的Java Web程序全部使用JSP实现，JSP页面既包含HTML标签，又包含Java业务逻辑代码，使得整个JSP页面臃肿而难于维护。

SUN公司为规范Java Web应用的开发，相继推出Model 1和Model 2模式。

在Model 1模式中引入JavaBean技术，使用Java Bean封装业务逻辑代码，使业务逻辑代码与JSP页面进行有效地分离，对于流程控制JavaBean却无能为力，直到Model 2模式引入Servlet技术才真正实现视图层、模型层与控制层的彻底分离，从此，Java Web应用开发进入了一个规范化的时代。

Struts 2作为Model 2模式的经典实现，具有组件的模块化，灵活性和重用性的优点，同时简化了基于MVC的Java Web应用程序的开发。

时代在发展，技术在进步，对各种复杂数据及业务逻辑的处理要求也在不断提高，MVC三层模型已不能完成满足高性能、低耦合框架结构的分层需求，此时，引入持久层概念，用于封装底层数据的持久化与访问操作。

在众多商用或开源的持久层中间件中，Hibernate 3.2由于其功能与性能的优越而成为Struts 2框架最理想的底层数据交换组件。

传统Java EE应用的开发效率低，应用服务器厂商对各种技术的支持并没有真正统一，导致Java EE的应用没有真正实现Write Once及Run Anywhere的承诺。

Spring 2.5作为开源的中间件，独立于各种应用服务器，甚至无须应用服务器的支持，也能提供应用服务器的功能，如声明式事务等。

Spring 2.5致力于Java EE应用的各层的解决方案，而不是仅仅专注于某一层的方案。

可以说Spring 2.5是企业应用开发的“一站式”选择，并贯穿表现层、业务层及持久层。

然而，Spring 2.5并不想取代那些已有的框架，而与它们无缝地整合。

以JSP 2技术为基础，整合Struts 2、Hibernate 3.2与Spring 2.5技术开发Java Web应用已成为主流的Java Web开发技术体系，被喻为Java EE开发领域的“黄金搭档”。

本书主要内容及其组织结构 本书以大型门户网站的开发为例，详细介绍了以JSP 2、Struts 2、Hibernate 3.2、Spring 2.5相结合的Java Web应用开发技术。

针对读者的不同学习阶段，共分技术篇、实践篇与部署篇三大部分进行循序渐进地讲解。

内容概要

本书详细介绍了以JSP2、Struts 2、Spring2.5、Hibernate 3.2, AJAX等主流Java EE技术进行优化组合开发高性能大型门户网站的实践经验。

针对读者的不同学习阶段,共分技术篇、实践篇与部署篇三大部分进行循序渐进的讲解。

第1-6章为技术篇,首先针对大型门户网站架构进行深入剖析,强调高性能,高负载大型门户网站开发对技术架构的严格要求,进而引出SSH 2组合技术架构,分别对Struts 2+Spring2.5+Hibernate 3.2架构的各组成部分进行深入阐述与实例讲解,图文并茂,让读者充分感受到SSH 2架构的优越性与重要性。

第7-12章为实践篇,紧紧围绕一个大型电子商务门户网站ePortal的各功能子系统进行理论联系实际的体验式讲解,技术重点、难点突出,手把手带领读者神游SSH 2大型门户网站的开发全过程,使读者在技术篇中学习的SSH 2技术得以进一步消化吸收。

第13-15章为部署篇,第13章理论联系实际,结合目前众多的SEO辅助工具对搜索引擎优化进行全面讲解;接下来第14章以业界主流的Red Hat Enterprise Linux 5(RHEL 5)作为服务器操作系统,继而针对RHEL 5的基本网络维护操作、JDK的安装、数据库服务器MySQL的安装与优化、Tomcat 6安装与优化、Apache2.2的安装与优化、Flash流媒体服务器,FTP服务器、SVN服务器及远程桌面实现等进行实例演示;第15章主要讲解负载均衡与服务器集群的技术实现,搭建高负载大型门户网站的服务器体系架构。

本书为高性能Java EE应用开发人员提供一整套实用的主流开发技术,融入了作者多年的高性能Java EE应用开发经验。

任何知识点均配以实例进行讲解,全程配有视频教程,大大缩短了读者的学习周期,降低了学习难度。

书中所有范例的源代码均保留完整的工程结构,方便读者朋友直接导入运行。

对于初学者而言,不失为一本好的入门教材;对于中级进阶者来说,本书也是一本难得的实践指南。

作者简介

刘斌，高级软件工程师Java EE资深讲师iChina互联在线Java EE架构师,于2007年出版《精通Java Web整合开发：JSP+AJAX+Struts+Hibernate》一书，从事Java EE企业级应用研究十余年，曾负责过多个大型电信、金融与房地产行业项目的系统架构与项目管理工作，积累了丰富的实践经验。对目前主流的Java EE开源框架颇有研究，尤其对轻量级Java EE的整合开发有独到的见解。

书籍目录

第1篇 技术篇	第1章 大型门户网站架构分析	1.1 大型门户网站与小型企业网站的区别	1.2 高性能、高负载门户网站架构剖析
	1.2.1 服务器操作系统的选择	1.2.2 DNS服务器BIND	1.2.3 Cache服务器Squid
	1.2.4 带负载均衡的HTTP服务器Apache	1.2.5 支持集群功能的Web服务器Tomcat	1.2.6 开源数据库服务器之骄子MySQL
	1.2.7 功能强大的FIV流媒体服务器RED5	1.3 门户网站开发指导思想	1.4 SSH 2组合框架—门户网站开发之首选
	1.4.1 MVC混血宠儿Struts 2	1.4.2 幕后的财政部长Spring 2.5	1.4.3 ORM中间件香馍馍Hibernate 3.2
	1.5 小结	第2章 MVC混血宠儿Struts 2	2.1 初识MVC新秀Struts 2
	2.1.1 MVC概述	2.1.2 Struts 2的MVC实现	2.1.3 Struts 2的基本组成
	2.1.4 Struts 2的常用类介绍	2.1.5 Struts 2的业务控制器Action实现	2.1.6 Struts 2的处理结果Result
	2.1.7 功能强大的表达式语言OGNL	2.1.8 Struts 2的运行流程分析—HelloWorld	2.1.9 Struts 2的异常处理
	2.2 Struts 2配置精要	2.2.1 web.xml中Struts 2的配置实现	2.2.2 Struts 2属性配置文件struts.properties详解
	2.2.3 Struts 2核心配置文件struts.xml详解	2.3 Struts 2应用开发实务	2.3.1 Struts 2应用开发环境的搭建
	2.3.2 Struts 2应用基本开发步骤—搭建ePortal应用	2.3.3 Struts 2中集成FCKeditor	2.3.4 Struts 2中集成DisplayTag
	2.4 Struts 2国际化实现	2.4.1 Web应用的中文本地化	2.4.2 Struts 2应用的国际化
	2.4.3 Struts 2国际化语言的动态切换	2.5 Struts 2的校验框架	2.5.1 在Action中实现手动校验
	2.5.2 调用校验框架进行自动校验	2.5.3 自定义国际化Struts 2校验错误消息	2.5.4 Struts 2的自带校验器
	2.6 Struts 2的拦截器	2.6.1 Struts 2内建拦截器介绍	2.6.2 定义和使用拦截器栈
	2.6.3 使用自定义拦截器	2.7 小结	第3章 Struts 2标签库详解与AJAX技术应用
	3.1 Struts 2主题与模板	3.2 Struts 2控制标签详解	3.2.1 if/elseif/else标签详解
	3.2.2 iterator标签详解	3.2.3 append标签详解	3.2.4 generator标签详解
	3.2.5 merge标签详解	3.2.6 subset标签详解	3.2.7 sort标签详解
	3.3 Struts 2数据标签详解	3.3.1 action标签详解	3.3.2 bean标签详解
	3.3.3 date标签详解	3.3.4 debug标签详解	3.3.5 i18n标签详解
	3.3.6 include标签详解	3.3.7 param标签详解	3.3.8 push标签详解
	3.3.9 set标签详解	3.3.10 text标签详解	3.3.11 url标签详解
	3.3.12 property标签详解	3.4 Struts 2表单标签详解	3.4.1 checkbox标签详解
	3.4.2 checkboxlist标签详解	3.4.3 combobox标签详解	3.4.4 doubleselect标签详解
	3.4.5 file标签详解	3.4.6 form标签详解	3.4.7 hidden标签详解
	3.4.8 select标签详解	3.4.9 optgroup标签详解	3.4.10 optiontransferselect标签详解
	3.4.11 password标签详解	3.4.12 radio标签详解	3.4.13 reset与submit标签详解
	3.4.14 textarea标签详解	3.4.15 textfield标签详解	3.4.16 token标签详解
	3.4.17 updownselect标签详解	3.5 Struts 2 AJAX标签详解	3.5.1 a与submit标签详解
	3.5.2 autocompleter标签详解	3.5.3 datetimepicker标签详解	3.5.4 div标签详解
	3.5.5 head标签详解	3.5.6 tabbedPanel标签详解	3.6 Struts 2其他标签详解
	3.6.1 actionerror与actionmessage标签详解	3.6.2 fielderror标签详解	3.6.3 tree与treenode标签详解
	3.7 Struts 2全力支持AJAX	3.7.1 Struts 2中原始AJAX技术应用	3.7.2 基于DWR的AJAX实现
	3.7.3 基于DWR的AJAX表单验证	3.7.4 基于JSON插件的AJAX实现	3.8 小结
	第4章 ORM中间件名流Hibernate 3接管持久层	4.1 ORM简介	4.1.1 持久化与持久层
	4.1.2 JDBC劣势	4.1.3 实体域模型与关系数据模型	4.1.4 ORM中间件
	4.2 Hibernate简介	4.2.1 Hibernate体系结构	4.2.2 Hibernate API简介
	4.2.3 配置Hibernate	4.2.4 Hibernate的映射配置文件	4.2.5 体验Hibernate (会员管理)
	4.3 Hibernate的映射机制	4.3.1 Hibernate的基本映射数据类型	4.3.2 Hibernate的主键映射
	4.3.3 Hibernate的实体映射	4.3.4 映射一对一关联关系	4.3.5 映射多对一单向关联关系
	4.3.6 映射一对多双向关联关系	4.3.7 映射一对多双向自身关联关系	4.3.8 映射多对多单向关联关系
	4.3.9 映射多对多双向关联关系	4.3.10 映射组成关系	4.3.11 映射继承关系
	4.3.12 Hibernate的集合映射	4.4 小结	第5章 Hibernate 3持久化技术实践与性能优化
	5.1 操作持久化对象的常用方法介绍	5.1.1 使用Session的beginTransaction()方法	5.1.2 使用Session的close()方法
	5.1.3		

使用Session的connection()方法	5.1.4 使用Session的delete()方法	5.1.5 使用Session的get()方法
5.1.6 使用Session的load()方法	5.1.7 使用Session的save()方法	5.1.8 使用Session的update()方法
5.1.9 使用Session的saveOrUpdate()方法	5.1.10 使用Hibernate的isInitialized()与initialize()方法	5.2 Hibernate的检索策略
5.1.11 持久化对象间的级联操作	5.3 HQL查询方式	5.2.1 立即检索
5.2.2 延迟检索	5.3.1 基本查询	5.2.3 迫切左外连接检索
5.3.2 条件查询	5.3.5 子查询	5.3.3 分页查询
5.3.4 连接查询	5.3.6 动态实例化查询结果	5.4.1 基本查询
5.3.5 子查询	5.4.2 QBE查询	5.4.4 复合查询
5.4.1 基本查询	5.4.3 分页查询	5.4.5 离线查询
5.4.2 QBE查询	5.6 Hibernate的批量处理	5.5 本地SQL查询
5.4.3 分页查询	5.6.1 批量插入	5.6.2 批量更新
5.6.1 批量插入	5.6.3 批量删除	5.7 Hibernate的事务管理
5.6.2 批量更新	5.7.1 事务边界声明	5.7.2 并发控制
5.6.3 批量删除	5.8 Hibernate的缓存机制	5.7.3 悲观锁
5.7.1 事务边界声明	5.8.1 Hibernate的缓存分类	5.7.4 乐观锁
5.7.2 并发控制	5.8.2 Hibernate的缓存范围	5.8.3 Hibernate的缓存管理
5.7.3 悲观锁	5.8.4 Hibernate二级缓存的并发访问策略	5.8.5 Hibernate的二级缓存配置
5.7.4 乐观锁	5.9 Hibernate应用的性能优化	5.10 多数据源的应用
5.8.1 Hibernate的缓存分类	5.11 JDBC的应用	5.12 Hibernate调用存储过程
5.8.2 Hibernate的缓存范围	5.13 XML数据持久化	5.14 小结
5.8.3 Hibernate的缓存管理	第6章 充分利用Spring 2.5的IoC利器统管Bean世界	6.1 Java程序员的春天厚礼—Spring 2.5
5.8.4 Hibernate二级缓存的并发访问策略	6.1.1 爱上Spring 2.5的十大理由	6.1.2 一睹Spring 2.5芳容
5.8.5 Hibernate的二级缓存配置	6.1.2 一睹Spring 2.5芳容	6.1.3 Spring 2.5拿手戏——控制反转与依赖注入
5.9 Hibernate应用的性能优化	6.1.4 何为“面向切面编程AOP”	6.1.5 Spring 2.5圣经——面向接口编程
5.10 多数据源的应用	6.1.6 开始Spring 2.5旅程—Hello World	6.2 Spring 2.5核心技术
5.11 JDBC的应用	6.2.1 Bean工厂之BeanFactory介绍	6.2.2 实用的Bean工厂ApplicationContext
5.12 Hibernate调用存储过程	6.2.2 实用的Bean工厂ApplicationContext	6.2.3 Spring 2.5配置文件详解
5.13 XML数据持久化	6.2.3 Spring 2.5配置文件详解	6.2.4 Bean的作用域与生命周期
5.14 小结	6.2.4 Bean的作用域与生命周期	6.2.5 基于XML方式的Bean装配
第6章 充分利用Spring 2.5的IoC利器统管Bean世界	6.2.5 基于XML方式的Bean装配	6.2.6 基于Annotation方式的Bean装配
6.1 Java程序员的春天厚礼—Spring 2.5	6.2.6 基于Annotation方式的Bean装配	6.2.7 基于XML方式的AOP实现
6.1.1 爱上Spring 2.5的十大理由	6.2.7 基于XML方式的AOP实现	6.2.8 基于Annotation方式的AOP实现
6.1.2 一睹Spring 2.5芳容	6.2.8 基于Annotation方式的AOP实现	6.3 Spring 2.5事务管理机制
6.1.3 Spring 2.5拿手戏——控制反转与依赖注入	6.3 Spring 2.5事务管理机制	6.3.1 Spring 2.5的声明式事务管理
6.1.4 何为“面向切面编程AOP”	6.3.1 Spring 2.5的声明式事务管理	6.3.2 基于XML方式的事务管理配置
6.1.5 Spring 2.5圣经——面向接口编程	6.3.2 基于XML方式的事务管理配置	6.3.3 基于Annotation方式的事务管理配置
6.1.6 开始Spring 2.5旅程—Hello World	6.3.3 基于Annotation方式的事务管理配置	6.4 “桃园三结义”——SSH 2组合开发框架始成
6.2 Spring 2.5核心技术	6.4 “桃园三结义”——SSH 2组合开发框架始成	6.4.1 Spring 2.5集成ORM中间件Hibernate 3.2
6.2.1 Bean工厂之BeanFactory介绍	6.4.1 Spring 2.5集成ORM中间件Hibernate 3.2	6.4.2 Spring 2.5集成MVC框架Struts 2
6.2.2 实用的Bean工厂ApplicationContext	6.4.2 Spring 2.5集成MVC框架Struts 2	6.4.3 SSH 2组合框架的基本开发步骤—ePortal启程
6.2.3 Spring 2.5配置文件详解	6.4.3 SSH 2组合框架的基本开发步骤—ePortal启程	6.5 小结 第2篇 实践篇
6.2.4 Bean的作用域与生命周期	6.5 小结 第2篇 实践篇	第7章 SSH 2热身——构建新闻发布系统
6.2.5 基于XML方式的Bean装配	第7章 SSH 2热身——构建新闻发布系统	7.1 门户网站新闻资讯基本分类
6.2.6 基于Annotation方式的Bean装配	7.1 门户网站新闻资讯基本分类	7.2 新闻发布系统需求分析
6.2.7 基于XML方式的AOP实现	7.2 新闻发布系统需求分析	7.3 新闻发布系统功能设计
6.2.8 基于Annotation方式的AOP实现	7.3 新闻发布系统功能设计	7.4 新闻栏目后台管理
6.3 Spring 2.5事务管理机制	7.4 新闻栏目后台管理	7.4.1 展示层JSP 2+Struts 2 标签库实现
6.3.1 Spring 2.5的声明式事务管理	7.4.1 展示层JSP 2+Struts 2 标签库实现	7.4.2 控制层Struts2实现
6.3.2 基于XML方式的事务管理配置	7.4.2 控制层Struts2实现	7.4.3 持久层Hibernate3实现
6.3.3 基于Annotation方式的事务管理配置	7.4.3 持久层Hibernate3实现	7.4.4 Spring2接管业务逻辑层
6.4 “桃园三结义”——SSH 2组合开发框架始成	7.4.4 Spring2接管业务逻辑层	7.5 新闻内容后台管理
6.4.1 Spring 2.5集成ORM中间件Hibernate 3.2	7.5 新闻内容后台管理	7.5.1 展示层JSP2+Struts2标签库实现
6.4.2 Spring 2.5集成MVC框架Struts 2	7.5.1 展示层JSP2+Struts2标签库实现	7.5.2 控制层Struts 2实现
6.4.3 SSH 2组合框架的基本开发步骤—ePortal启程	7.5.2 控制层Struts 2实现	7.5.3 持久层Hibernate3实现
6.5 小结 第2篇 实践篇	7.5.3 持久层Hibernate3实现	7.5.4 Spring2接管业务逻辑层
第7章 SSH 2热身——构建新闻发布系统	7.5.4 Spring2接管业务逻辑层	7.6 新闻资讯前台展现
7.1 门户网站新闻资讯基本分类	7.6 新闻资讯前台展现	7.6.1 文本新闻列表自定义标签类TextnewsTag
7.2 新闻发布系统需求分析	7.6.1 文本新闻列表自定义标签类TextnewsTag	7.6.2 文本新闻列表分页控制自定义标签类TextnewsPageTag
7.3 新闻发布系统功能设计	7.6.2 文本新闻列表分页控制自定义标签类TextnewsPageTag	7.6.3 图片新闻列表自定义标签类PicnewsTag
7.4 新闻栏目后台管理	7.6.3 图片新闻列表自定义标签类PicnewsTag	7.6.4 幻灯片新闻列表自定义标签类SlidenewsTag
7.4.1 展示层JSP 2+Struts 2 标签库实现	7.6.4 幻灯片新闻列表自定义标签类SlidenewsTag	7.6.5 Struts2与AJAX完全结合实现点击数加一
7.4.2 控制层Struts2实现	7.6.5 Struts2与AJAX完全结合实现点击数加一	7.7 基于HttpClient构建通用静态页面发布引擎
7.4.3 持久层Hibernate3实现	7.7 基于HttpClient构建通用静态页面发布引擎	7.7.1 优秀的模拟浏览器HttpClient
7.4.4 Spring2接管业务逻辑层	7.7.1 优秀的模拟浏览器HttpClient	7.7.2 构建通用静态页面发布引擎
7.5 新闻内容后台管理	7.7.2 构建通用静态页面发布引擎	7.8 小结
7.5.1 展示层JSP2+Struts2标签库实现	7.8 小结	第8章 通用新闻采集系统
7.5.2 控制层Struts 2实现	第8章 通用新闻采集系统	8.1 新闻采集系统需求分析
7.5.3 持久层Hibernate3实现	8.1 新闻采集系统需求分析	8.2 新闻采集系统功能设计
7.5.4 Spring2接管业务逻辑层	8.2 新闻采集系统功能设计	8.3 使用正则表达式与HTMLParser采集新闻内容
7.6 新闻资讯前台展现	8.3 使用正则表达式与HTMLParser采集新闻内容	8.3.1 正则表达式基本知识
7.6.1 文本新闻列表自定义标签类TextnewsTag	8.3.1 正则表达式基本知识	8.3.2 体验HTMLParser
7.6.2 文本新闻列表分页控制自定义标签类TextnewsPageTag	8.3.2 体验HTMLParser	8.4 量身定制采集规则
7.6.3 图片新闻列表自定义标签类PicnewsTag	8.4 量身定制采集规则	8.4.1 展示层JSP 2+Struts 2 标签库实现
7.6.4 幻灯片新闻列表自定义标签类SlidenewsTag	8.4.1 展示层JSP 2+Struts 2 标签库实现	8.4.2 控制层Struts 2实现
7.6.5 Struts2与AJAX完全结合实现点击数加一	8.4.2 控制层Struts 2实现	8.4.3 持久层Hibernate3实现
7.7 基于HttpClient构建通用静态页面发布引擎	8.4.3 持久层Hibernate3实现	8.4.4 Spring 2接管业务逻辑层
7.7.1 优秀的模拟浏览器HttpClient	8.4.4 Spring 2接管业务逻辑层	8.5 新闻采集功能实现
7.7.2 构建通用静态页面发布引擎	8.5 新闻采集功能实现	8.6 小结
7.8 小结	8.6 小结	第9章 会员注册及会员管理系统
第8章 通用新闻采集系统	第9章 会员注册及会员管理系统	9.1 会员管理系统需求分析
8.1 新闻采集系统需求分析	9.1 会员管理系统需求分析	9.2 会员管理系统功能设计
8.2 新闻采集系统功能设计	9.2 会员管理系统功能设计	9.3 会员级别后台管理
8.3 使用正则表达式与HTMLParser采集新闻内容	9.3 会员级别后台管理	9.3.1 展示层JSP 2+Struts 2 标签库实现
8.3.1 正则表达式基本知识	9.3.1 展示层JSP 2+Struts 2 标签库实现	9.3.2 持久层Hibernate 3实现
8.3.2 体验HTMLParser	9.3.2 持久层Hibernate 3实现	9.3.3 持久层Hibernate 3实现
8.4 量身定制采集规则	9.3.3 持久层Hibernate 3实现	9.3.4 Spring 2接管业务逻辑层
8.4.1 展示层JSP 2+Struts 2 标签库实现	9.3.4 Spring 2接管业务逻辑层	9.4 注册会员后台管理
8.4.2 控制层Struts 2实现	9.4 注册会员后台管理	9.4.1 展示层JSP 2+Struts 2 标签库实现
8.4.3 持久层Hibernate3实现	9.4.1 展示层JSP 2+Struts 2 标签库实现	9.4.2 控制层Struts2实现
8.4.4 Spring 2接管业务逻辑层	9.4.2 控制层Struts2实现	9.4.3 持久层Hibernate3实现
8.5 新闻采集功能实现	9.4.3 持久层Hibernate3实现	9.4.4 Spring 2接管业务逻辑层
8.6 小结	9.4.4 Spring 2接管业务逻辑层	9.5 前台会员注册与登录
第9章 会员注册及会员管理系统	9.5 前台会员注册与登录	9.6 小结
9.1 会员管理系统需求分析	9.6 小结	第10章 举一反三—实现商品分类管理
9.2 会员管理系统功能设计	第10章 举一反三—实现商品分类管理	10.1 商品管理系统需求分析
9.3 会员级别后台管理	10.1 商品管理系统需求分析	10.2 商品管理系统功能设计
9.3.1 展示层JSP 2+Struts 2 标签库实现	10.2 商品管理系统功能设计	10.3 商品类别后台管理
9.3.2 持久层Hibernate 3实现	10.3 商品类别后台管理	10.3.1 展示层JSP 2+Struts 2 标签库实现
9.3.3 持久层Hibernate 3实现	10.3.1 展示层JSP 2+Struts 2 标签库实现	10.3.2 控制层Struts 2实现
9.3.4 Spring 2接管业务逻辑层	10.3.2 控制层Struts 2实现	10.3.3 持久层Hibernate3实现
9.4 注册会员后台管理	10.3.3 持久层Hibernate3实现	10.3.4 Spring2接管业务逻辑层
9.4.1 展示层JSP 2+Struts 2 标签库实现	10.3.4 Spring2接管业务逻辑层	10.4 商品后台管理
9.4.2 控制层Struts2实现	10.4 商品后台管理	10.4.1 展示层JSP 2+Struts 2 标签库实现
9.4.3 持久层Hibernate3实现	10.4.1 展示层JSP 2+Struts 2 标签库实现	10.4.2 控制层Struts 2实现
9.4.4 Spring 2接管业务逻辑层	10.4.2 控制层Struts 2实现	10.4.3 持久层Hibernate3实现
9.5 前台会员注册与登录	10.4.3 持久层Hibernate3实现	10.4.4 Spring 2接管业务逻辑层
9.6 小结	10.4.4 Spring 2接管业务逻辑层	10.5 小结
第10章 举一反三—实现商品分类管理	10.5 小结	第11章 温故知新—打造购物车与订单管理系统
10.1 商品管理系统需求分析	第11章 温故知新—打造购物车与订单管理系统	11.1 购物车与订单管理系统
10.2 商品管理系统功能设计	11.1 购物车与订单管理系统	
10.3 商品类别后台管理		
10.3.1 展示层JSP 2+Struts 2 标签库实现		
10.3.2 控制层Struts 2实现		
10.3.3 持久层Hibernate3实现		
10.3.4 Spring2接管业务逻辑层		
10.4 商品后台管理		
10.4.1 展示层JSP 2+Struts 2 标签库实现		
10.4.2 控制层Struts 2实现		
10.4.3 持久层Hibernate3实现		
10.4.4 Spring 2接管业务逻辑层		
10.5 小结		

<<大型门户网站是这样炼成的! >

需求分析	11.2 购物车及订单管理系统功能设计	11.3 购物车自助管理	11.3.1 展示
层JSP2+Struts2标签库实现	11.3.2 控制层Struts 2实现	11.3.3 持久层Hibernate3实现	
11.3.4 Spring2接管业务逻辑层	11.4 订单管理	11.4.1 展示层JSP 2+Struts 2标签库实现	
11.4.2 控制层Struts 2实现	11.4.3 持久层Hibernate 3实现	11.4.4 Spring 2接管业务逻辑层	
11.5 小结	第12章 站务必备—系统用户管理与流量统计	12.1 系统用户管理与流量统计需求分析	
12.2 系统用户管理与流量统计功能设计	12.3 系统用户管理	12.3.1 展示层JSP 2+Struts 2标	
签库实现	12.3.2 控制层Struts 2实现	12.3.3 持久层Hibernate3实现	12.3.4 Spring2接管业
务逻辑层	12.4 流量统计分析	12.4.1 展示层JSP 2+Struts 2标签库实现	12.4.2 控制层Struts
2实现	12.4.3 持久层Hibernate3实现	12.4.4 Spring2接管业务逻辑层	12.5 小结 第3篇 部署
篇 第13章 门户网站的搜索引擎优化—SEO技术	13.1 为什么要进行搜索引擎优化	13.1.1 搜	
索引擎概述	13.1.2 网络营销催生SEO	13.1.3 SEO让你的网站脱颖而出	13.2 搜索引擎到
底喜欢什么样的网站	13.2.1 影响网站自然排名的主要因素	13.2.2 评价网站优劣的重要指标	
13.2.3 正确看待SEO	13.2.4 远离Google的“沙盒”	13.3 搜索引擎优化最佳实践	
13.3.1 如何快速被搜索引擎收录	13.3.2 建立合理的网站结构与网站地图的应用	13.3.3 设计	
良好的网页结构	13.3.4 SEO基本功——关键词的选用	13.3.5 域名与空间对SEO的影响	
13.3.6 搜索引擎偏爱“勤快”的网站	13.4 搜索引擎优化辅助工具大本营	13.4.1 关键字选	
词辅助工具	13.4.2 关键字排名查询	13.4.3 关键字密度查询	13.4.4 热门关键字及搜索
量查询	13.4.5 PR值查询	13.4.6 Alexa世界排名查询	13.4.7 搜索引擎收录页面数查询
13.4.8 搜索引擎反向链接查询	13.4.9 网站地图制作工具	13.4.10 百度指数查询	
13.4.11 网页中无效链接检测工具	13.5 小结	第14章 基于RHEL 5的服务器架构	14.1 Red Hat
Enterprise Linux 5安装与优化	14.2 Linux客户端工具Secure SSH安装与应用	14.3 RHEL5服务器维	
护的基本操作	14.3.1 系统维护相关的Linux命令	14.3.2 文件操作相关的Linux命令	
14.3.3 网络维护相关的Linux命令	14.3.4 日志查看相关的Linux命令	14.3.5 用户管理相关	
的Linux命令	14.3.6 应用软件安装相关的Linux命令	14.4 RHEL5中的JDK1.6安装	14.5
RHEL5中的MySQL5安装与性能优化	14.6 基于vsftpd搭建FTP服务器	14.7 RHEL5中的Tomcat6安	
装与性能优化	14.8 RHEL5中的Apache2.2安装与性能优化	14.9 Apache2.2配置文件httpd.conf详	
解	14.10 使用subversion搭建SVN版本控制服务器	14.11 利用Red5搭建Flash流媒体服务器	
14.12 利用vncserver服务实现远程桌面	14.13 小结	第15章 负载均衡与服务器集群	15.1 Web
服务器集群概述	15.2 搭建Apache2.2+Tomcat6分工协作服务器架构	15.3 构建Apache2.2负载均衡	
与Tomcat6集群	15.4 搭建主从结构的MySQL数据库集群	15.5 基于MySQL Proxy的负载均衡与读	
写分离	15.6 小结		

章节摘录

第1篇 技术篇 第1章 大型门户网站架构分析 1.2 高性能、高负载门户网站架构剖析

1.2.4 带负载均衡的HTTP服务器Apache 负载均衡 (Load Balance) 建立在现有网络结构之上, 它提供了一种廉价、有效、透明的方法, 来扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

负载均衡有两方面的含义: 首先, 大量的并发访问或数据流量分担到多台结点设备上分别处理, 减少用户等待响应的时间; 其次, 单个重负载的运算分担到多台结点设备上做并行处理, 每个结点设备处理结束后, 将结果汇总, 返回给用户, 系统处理能力得到大幅度提高。

负载均衡从其应用的地理结构上, 分为本地负载均衡和全局负载均衡。

本地负载均衡是指对本地的服务器群做负载均衡, 全局负载均衡是指在不同地理位置对不同网络结构的服务器群间做负载均衡。

本地负载均衡能有效地解决数据流量过大、网络负荷过重的问题, 并且不需花费昂贵开支购置性能卓越的服务器, 可充分利用现有设备, 避免服务器单点故障造成数据流量的损失。

有灵活多样的均衡策略, 可把数据流量合理地分配给服务器群内的服务器来共同负担。

即使是再给现有服务器扩充升级, 也只是简单地增加一个新的服务器到服务群中, 而不需改变现有网络结构、停止现有的服务。

全局负载均衡, 主要用于在一个多区域拥有自己服务器的站点, 为了使全球用户只以一个IP地址或域名就能访问到离自己最近的服务器, 从而获得最快的访问速度。

<<大型门户网站是这样炼成的！>>

编辑推荐

构建大型门户网站，难！构建高性能的大型门户网站，更难！这门综合技术，一直秘而不宣，但它们是如何炼成的呢？《大型门户网站是这样炼成的！

（Struts 2+Spring 2+Hibernate 3）》深度剖析JavaEE轻量级组合框架SSH2的应用实践，全新展示大型门户网站的架构、开发、部署，以及SEO与性能提升等实践经验。

随书附赠DVD光盘，内含《大型门户网站是这样炼成的！

（Struts 2+Spring 2+Hibernate 3）》全部源代码，以及12小时视频教程，生动讲解相关核心技术，更拓展《大型门户网站是这样炼成的！

（Struts 2+Spring 2+Hibernate 3）》纸上内容。

《大型门户网站是这样炼成的！

（Struts 2+Spring 2+Hibernate 3）》以大型门户网站的开发为例，详细介绍了以JSP 2、Struts 2、Hibernate 3.2、Spring 2.5相结合的Java Web应用开发技术。

技术篇 主要针对高性能大型门户网站开发的SSH 2组合框架进行讲解，另外对门户网站的整个体系结构、服务器操作系统的选用、门户网站的开发指导思想及各种服务器软件的应用也进行全面介绍，使读者朋友对门户网站的开发有一个较全面的了解。

实践篇 以一个大型的电子商务门户网站项目ePortal为综合实例，淋漓尽致地展示了SSH 2组合框架技术的具体应用细节，使读者对在技术篇中学习的SSH 2技术得以进一步消化吸收。

部署篇 重点讲解SEO搜索引擎优化及基于RHEL5的服务器架构技术。

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>