

杭州电子科技大学

毕业设计（论文）文献综述

毕业设计(论文)题目	基于 SpringBoot 的响应式技术博客 的设计和实现
文献综述题目	SpringBoot 颠覆 JavaEE 的开发模式
学 院	计算机学院
专 业	软件工程
姓 名	朱文赵
班 级	14108413
学 号	14108337
指导教师	马虹

SpringBoot 颠覆 JavaEE 的开发模式

一、前言

在脚本语言和敏捷开发大行其道的时代，JavaEE 的开发显得尤为笨重，让人误解 JavaEE 开发就应该如此。Spring 在提升 JavaEE 开发效率的脚步上从未停止过，而 Spring Boot 的推出是具有天赋和划时代意义的^[1]。本文介绍了 SpringBoot 和分析其与 JavaEE 开发体系的不同。分析以 SpringBoot 为核心的开发框架组成。最后分析部署、测试、监控和管理生产环境等问题。

二、主题

1. SpringBoot 与传统 JavaEE 项目开发模式比较。

JavaWeb 应用开发从最早的 Servlet 技术开始已经经历了很多变化，在这段历史中产生了独立的一整套的生态系统和大量的开源项目^[2]。EJB2.0 改变了很多，但也还是保留了权威、官方和学术化的整体风格。在这个时候 Spring 等大批轻量级框架开始产生，发展到今天已经和 JavaEE 的开发模式形成有明显区别的一派，真正让我们看到开源社区改变 JavaEE 的应该是 JAP，它的标准明显倾向于 Hibernate。Java 被 Oracle 收购后对 JavaEE 不断发展但开发模式没有很大的发展。SpringBoot 是在 Spring 的大框架下，离不开两项关键技术依赖注入和面向切面编程 AOP。在 JDK5 引入的 Annotation 出现后 Spring 大量采用这种配置方式，但之前都是采用配置 xml 文件的形式。xml 文件在 webservice 流行的时期被推崇的很高。开发人员渐渐发现 xml 文件越来越多，书写困难(可以对比 YAML 文件的书写)，调试困难等等^[3]。所以 Spring 当初被人的批评很多都是因为 xml 配置过多。而 Spring Boot 的改变不是仅仅采用了更好的配置文件格式那么简单。SpringBoot 是采用了最初在 Maven 中看到的“约定大于配置”(Convention over configuration)的思想^[4]，也叫惯例优先原则。简单地解释就是绝大多数配置信息按约定采用缺省配制，用户需要的个性配置采用 properties 文件或 yaml 文件的格式写在统一的文件中。其他在开发中的配置信息要么采用 Annotation 要么采用配置类形式。也就是说 SpringBoot 可以达到免 xml 文件配置。这点区别于传统项目开发的特点也是给开发效能提升带来巨大效果的。

Spring Boot 的运行方式是 main 方法。另外这个项目是 Javaweb 项目，它必然需要 web 容器，但这不是 war 包形式部署。实际上 SpringBoot 改变了传统开发模式，将其开发的 javaweb 项目内部内嵌了一个 Tomcat,而关键是最终的打包形式是一个包含了所有项目文件和内嵌、web 容器的 jar 包。这个 jar 包的打包形式为将来部署到 Docker 上

提供了良好的条件，这个后面还会提到。SpringBoot 使 Javaweb 项目开发变得轻松很多^[5]，当然它的贡献也不止以上这些，当项目真正采用它来开发一般不会再退回原来的开发方式了。

2. SpringBoot 应用系统开发模板架构设计从前台到后台的顺序进行分析。

从页面来看，现在的主流已经淘汰了 jsp 技术。前台常使用模板引擎，主要有 FreeMarker,它是一个用 Java 语言编写的模板引擎，基于模板来生成文本输出。还有 Thymeleaf,它和 FreeMarker 的原理类似，可以达到页面图形设计和应用逻辑的分离。但 SpringBoot 推荐使用 Thymeleaf 正原因首先是它可以和 SpringMVC 很好结合，其次和其他模板引擎相比它的模板可以直接用浏览器正确显示，这也称为自然的模板技术。但其使用的 DOM 解析所以不适合处理大 XML 文件。在前端开发还常用到 Bootstrap、AngularJS、JQuery 等。接下来在浏览器数据传输上采用 Json 的格式 RESTful 传输。在数据到达服务器后接受请求的是 SpringMVC 框架，这个框架已经取代 struts2 成为控制层绝对的主流了。到持久层框架现在的主流技术有 Hibernate、Mybatis、JPA 各有优缺点不再赘述。SpringBoot 推荐使用 Spring Data+JPA 的方式^[6]，当然引擎常用 Hibernate,数据库使用 MySQL 开发工具推荐用 IntelliJ IDEA。

3. SpringBoot 部署和测试带来的变革。

SpringBoot 项目能够很好地部署于 Docker 类的虚拟容器中，Docker 由 2013 年 DotCloud 公司发起，开源后快速发展。已经形成以 Docker 为中心的生态体系。Docker 加入了 Linux 基金会，遵循 Apache2.0 协议。Docker 的受欢迎程度很高，应用该项技术的大型互联网公司几乎占据主流市场的三分之一以上。SpringBoot 也针对 Docker 技术做了很多设计和优化以适应容器技术。Docker 引擎的基础组件是 Linuxcontainers(LXC)。各种工具的丰富使 Docker 应用迅速普及，正如 Solomon Hykes 所说:Docker 在正确的地点、正确的时间顺应了正确的趋势——即高效地构建应用^[7]。隔离是 Docker 的重点，正如 JVM 隔离了硬件资源环境影响一样，Docker 将容器外环境的改变变得不那么重要了，程序的移植变得象拷贝文件一样简单了。这给运维带来了极大的方便。利用 SpringBoot Maven plugin 将 Maven 引入 SpringBoot 项目是一个关键，Spotify 的 docker-maven-plugin 插件可以用来构建 Maven 的 Docker 镜像。可以编写 Dockerfile 来指定特定的镜像层。之后可以将镜像推送到 DockerHub 上，DockerHub 是一个类似 GitHub 原理的镜像托管平台。在项目开发阶段最好的做法是使用持续集成的做法，即使用 Jenkins 来结合 Git Hooks，当代码 Push 到 Git 服务器后，Jenkins 自动去执行 shell 脚本文件 Pull 的代码，又 Maven 变异打包生成最新的镜像文件。最后再 push 到 DockerHub 上。装有 Docker 的服务器从 DockerHub 上的到镜像

运行该镜像的所属容器从而完成整个部署过程。另外由于在 SpringBoot 中的模板引擎默认开启缓存，利用 SpringLoaded 实现类文件修改热部署，利用 JRebel 可以实现开发热部署，这样会极大提高开发调测效率。

SpringBoot 在测试方面也带了很多的改变^[8]。首先，在 SpringBoot 项目中按照其一惯的 starter 做法在项目中加入 spring-boot-starter-test 的依赖，其他的关于测试的依赖就不用再管理了。根据 SpringBoot 的最佳实践在项目 src/test/java 目录下添加测试类。其次，测试类 SpringBoot 使用 @springapplicationconfiguration 代替了 @contextConfiguration 配置 SpringBoot 的应用上下文，还有两个常用的 Annotation 是 @RunWith(SpringJUnit4ClassRunner.class)和 @WebApplication 和 @transactional 保证测试操作的回滚，具体的测试方法就完全按照 Junit 的方式做了。在 controller 层测试类为了 MockMvc 而注入了 WebApplicationContext^[9]。这里充分利用了 SpringMVC 的测试功能。测试的过程可以总结归纳为以下过程，为项目准备测试的环境；通过 MockMvc 来模拟执行 http 请求；增加验证的 JUnit Assert 还有处理结果 Assert。收到 MvcResult 后进行自定义 Assert 又从新进行下一步的异步请求；最后卸载掉测试用的 MockMvc 环境，完成测试。总之 SpringBoot 利用和 SpringMVC 的关系将测试变得简单实用，为后期自动化集成测试奠定良好的基础^[10]。

4. SpringBoot 监控和管理生产环境。

SpringBoot 给 web 系统开发带来的改变中一个重要的方面就是监控和管理生产环境^[11]。如何实时监控应用系统信息的各项功能是否健康，就要通过系统各方面的性能安全指标数值来评估。这样可以在系统变得不稳定的时期就发现或解决问题，而不是系统功能停止时刻才发现，从而避免业务遭受打击。大规模互联网应用一定会有专门的网管系统，但应用系统内部情况就不会监控得那么细致了。Spring 可以监控到什么信息呢，如：当前应用的所有配置、当前应用中所有的 Bean 的信息、所有配置属性、线程状态信息、当前环境信息、应用健康状况、当前应用的各项指标信息、所有 @RequestMapping 映射的路径、显示追踪信息以及所有 EndPoint 列表^[12]。可以看到内容非常丰富而且实用。这些监控信息通过监控和管理端点的形式提供给开发者。而前提是在 POM 文件中加入 spring-boot-starter-actuator 这个依赖。在创建项目时可以选择 Actuator、web 和 HATEOAS 这几个模块。当项目部署完成可以通过 http URL 地址访问到这些监控信息，如：http://localhost:8080/actuator 可以看到所有 EndPoint 列表；http://localhost:8080/beans 可以看到所有 Beans 列表等。甚至还可以自定义端点对一些业务指标进行监控，这是任何市场盒装系统监控产品也做不到的^[13]。另外 SpringBoot 在 web 系统监控开发的提升开可以利用 JMX(Java Management Extensions，即 Java 管理扩展)进行^[14]。

三、结论

SpringBoot 对于 Spring 平台无疑是巨大的提升^[15]。无论在开发阶段还是部署测试阶段，甚至系统在线运行阶段，SpringBoot 都在发挥着他的作用。使 web 系统开发在多方面整体提升。随着微服务的推广，作为底层支持的 SpringBoot 正在改变 web 应用开发模式^[16]。

总之虽然 Spring Boot 给我们带来了类似于脚本语言开发的效率，但是 Spring Boot 里没有使用任何让我们意外的技术^[17]，完全是一个单纯的基于 Spring 的应用。如 Spring Boot 的自动配置是通过 Spring4.x 的@Conditional 注解来实现的。并且 Spring 技术栈中的 SpringCloud 已经成为新一代微服务框架，成为比国内 Dubbo 的更加成熟的微服务宠儿。同时还有 Spring Cloud Data Flow 等成熟框架，这些都将影响着将来的开发模式和开发效率，并且这将会是一种趋势，就像 Python 适合来做机器学习一样。因为他们都是相关领域中最适合该生态环境的方式。

四、参考文献

- [1] 张峰.应用 SpringBoot 改变 web 应用开发模式[J]. 科技创新与应用, 2017(23): 193-194.
- [2] 汪云飞. JavaEE 开发的颠覆者: Spring Boot 实战[M]. 北京: 电子工业出版社, 2016: 1-15.
- [3] Pivotal 团队. Spring Boot Reference Guide1.5.3.RELEASE[OL]. 2017.
- [4] 王永和, 张劲松, 邓安明, 周智勋. Spring Boot 研究和应用[J]. 信息通信, 2016(10): 91-94.
- [5] 杨家炜.基于 Spring Boot 的 web 设计与实现[J]. 轻工科技, 2016, 32(07): 86-89.
- [6] 郑彬彬. 基于微服务的 OJ 系统重构与优化[D]. 上海: 东华大学, 2017.
- [7] Bhimani J, Yang Z, Mi N, et al. Docker Container Scheduler for I/O Intensive Applications running on NVMe SSDs[J]. 2018, 2(99): 1-1.
- [8] Reddy K S P. Testing Spring Boot Applications[M]. Beginning Spring Boot 2. Apress, Berkeley, CA, 2017: 221-246.
- [9] Reddy K S P. Getting Started with Spring Boot[M]. Beginning Spring Boot 2. Apress, Berkeley, CA, 2017: 21-33.
- [10] Walls C. Spring Boot in action[M]. Manning Publications Co., 2016:21-34
- [11] Felipe Gutierrez. Spring Boot, Simplifying Everything[M]. Apress: 2014-06-15:02-06
- [12] Miller R A, Heitkamp E M. Spring boot: U.S. Patent Application 12/435,432[P]. 2010-11-11.

- [13]Balalaie A, Heydarnoori A, Jamshidi P. Microservices architecture enables devops: Migration to a cloud-native architecture[J]. IEEE Software, 2016, 33(3): 42-52.
- [14]Balalaie A, Heydarnoori A, Jamshidi P. Migrating to cloud-native architectures using microservices: an experience report[C].European Conference on Service-Oriented and Cloud Computing. Springer, Cham, 2015: 201-215.
- [15]Webb P, Syer D, Long J, et al. Spring boot reference guide[J]. Part IV. Spring Boot features, 2013, 10(3): 24-25.
- [16]Prieto Barreiro I, Varela F, Mandilara E. Monitoring of CERN's Data Interchange Protocol (DIP) System[J]. 2018, 3(33): 3-5.
- [17]Varela F, Gonzalez Corral M, Podgorski S, et al. MARS: Easing Maintenance and Interventions for CERN Controls[J]. 2018, 3(11): 3-4.

文献综述考核表

指导教师对文献综述的评语：

指导教师 _____（签名）

2018 年 3 月 16 日

建议成绩	
------	--

评阅小组或评阅人对文献综述的评语：

评阅小组负责人或评阅人 _____（签名）

2018 年 3 月 16 日

建议成绩	
------	--