1 绪论

1.1 课题研究背景及意义

当今社会是一个“信息爆炸”的社会，尤其是进入互联网时代后，人们每天接触 的信息和互联网产品远远大于互联网发展初期的年代[2]。随着中国互联网技术的 不断发展以及 WEB 应用开发成本的不断降低，人们对于 WEB 应用的需求不再满 足于通过缓慢的加载和复杂的操作获取资讯[3]，许多的创业者希望通过开发一个 WEB 应用实现自己创业梦想的时代已经无法获得投资人的青睐。而且，由于近几 年 WEB 信息泄露的新闻不断进入消费者的视野，用户对于 WEB 应用的安全性体 验也提出了更高的要求。

随着互联网技术进一步普及，基于模型-视图-控制器 (Model-View-Controller, MVC) 模式的 WEB 应用程序[4] 被广大开发者所使用，目前主流的 WEB 应用程序 均在使用此框架。使用 MVC 框架，可以将 WEB 应用进行清晰的分层开发，前端 开发人员主要负责页面的呈现方式和用户体验，后台人员则主要负责应用的逻辑 实现以及数据结构的搭建，极大的降低了研发成本。

随着信息化的进一步普及，数据库的使用越来越广泛，数据成为一个应用甚 至也个企业最重要的价值体现，越来越多的企业发展离不开数据库。因此数据库 的稳定性和安全性也称为很多企业研发的重点。通过设计数据的管理和使用机制， 在提高数据库使用效率的同时，保障数据的完整性和安全性是企业运维人员的在 运维过程中的重要任务。

除此之外，如何实现WEB应用的高可用性和分布式服务也是保障用户体验的 一个很重要的部分，通过设计基于应用和服务器的不同层级不同纬度的优化策略， 提升应用的高可用性也是在 WEB 应用开发中必须要注意的。

1.2 国内外研究现状

近几年，随着计算机技术和 Internet 的迅速发展，网络的信息量和访问量成

几何数级增长，网络拥塞和服务超载日益成为网络及其服务器必须面对的严峻问 题[5]。

2006 年新华网被黑事件、2010 年的百度域名劫持时间、2011 年的 CSDN 用户 数据泄漏事件以及 2015 年网易邮箱密码泄漏事件等事件无不说明系统优化的作用 和意义。

目前大部分互联网产品在开始发展的阶段都重视了产品的设计和功能，然而 在产品的稳定性和安全性方面却考虑不足，这在一定程度上增加了互联网产品在 推广过程中潜在的风险。因此，在开发互联网产品的同时，通过一系列的优化策 略，提升应用的体验，增强应用的稳定性和安全性必须得到充分的认识。

1.3 本文主要研究内容 1.3.1 主要工作

论文在满足应用正常运行的基础上，通过不同维度的优化实践研究基于 Spring MVC 架构 WEB 应用的系统优化策略。首先通过搭建基于 Spring MVC 的 WEB 测 试应用，在实际用户使用的过程中通过不断开发和调整系统的优化策略，提升应 用的用户体验和应用的稳定性，探索行之有效的系统优化策略。

本文的研究对象主要有以下内容:

(1) 对 Jenkins 持续集成环境研究。自动化部署和代码检测是保证基于 WEB 的 应用产品质量的一个重要环节，通过研究部署 Jenkins 集成测试环境，对编写的代 码进行版本控制、自动化构建和代码测试，研究持续集成方案对于 WEB 应用系统 优化策略的影响。

(2) 对 Couchbase 缓存机制进行研究。目前大多数的 WEB 应用对于缓存性能 的优化还没有足够的重视，通过开发针对 WEB 应用的 Couchbase 缓存系统，研究 Couchbase 的缓存机制测试有效的缓存机制对于系统性能提升和用户体验的影响， 探索可用的系统优化策略。

(3) 对 Docker 容器编排技术进行研究。随着用户的不断增加，单一节点的 WEB 应用或数据库应用已经无法满足用户的需求，如何快速的部署新的应用节点提升 应用性能成为开发者关注的问题。通过构建基于Docker容器编排技术的应用容器， 在新的服务器快速部署新的应用，并加入到应用集群中去，探索 Docker 机制对于 系统优化策略的影响。

(4) 对数据库的主从复制进行研究。随着应用的发展，数据逐渐成为应用最有 价值的部分，如何更好的保证数据的完整性以及安全性在应用的维护过程中日益 重要。通过研究数据库的主从复制和延时复制方案，探索数据库优化对于系统安 全性优化的策略。

(5) 对基于 API 的监控和应急措施方案进行研究。随着服务节点的不断增加， 服务的高可用和服务的健康性检查时 WEB 应用的运维人员在维护过程中必须要 注意的方面。通过开发基于 API 的服务器、服务监控系统以及基于 API 的应急处 理系统研究 API 操作对于系统的快速检测和快速恢复的影响。

(6) 其他方面，研究 WEB 应用的搭建过程、Tomcat 的相关配置、负载均衡以

及阿里云的相关配置研究 WEB 应用和服务器自身优化对于系统性能的影响。 1.3.2 论文目标

本论文致力于分析目前很多 WEB 应用开发过程中存在的问题和不完善的地 方，并探索 WEB 应用性能和服务器性能的优化方案。

在 Linux 平台上，使用 JAVA 语言和 MySQL 数据搭建一个基于 Spring MVC 架构的 WEB 应用，通过 Tomcat 实现应用的访问和测试，通过设计不同的优化方 案对 WEB 进行测试，研究在实际应用中有价值的优化策略。

系统性能优化策略研究主要包含应用层面、数据层面和服务器层面三个层面。 在应用层面，通过持续集成、代码审核实现应用的稳定性和安全性。在数据库层 面，通过多数据库主从复制、负载均衡和备份恢复等策略实现数据的高可用和稳 定性。在服务器层面，通过对服务的监控和服务器的监控、服务器间负载均衡的 配置、基于 API 的自动化 failover 等策略保障服务的正常运行和高负载应对。

1.4 论文组织结构

基于 Spring MVC 架构 WEB 应用的系统优化策略的研究是按照计划逐步完成 的，本人将此研究分为六个部分:

第一章:绪论。本章主要介绍 WEB 应用开发的现状和系统优化策略研究的意 义，明确了在 WEB 应用开发过程中对于应用和服务器进行优化的必要性和重要 性。介绍了本论文的主要工作内容、目标以及创新点。最后介绍了论文的编排结 构。

第二章:WEB 应用开发介绍。本章主要介绍了前端和后端的开发框架和开发 流程以及数据库的搭建过程。然后介绍了 Tomcat 的配置过程。之后介绍了 Jenkins 持续集成环境的搭建和使用过程。

第三章:应用性能优化介绍。本章主要介绍了在 WEB 应用性能优化方面册主 要方法和策略，主要包括 Couchbase 缓存优化，Tomcat 高性能 Apr 配置，Docker 分布式优化以及 SLB 负载均衡优化等方面。同时对应用的安全性和稳定性进行了 分析。

第四章:数据优化。本章主要介绍了 MySQL 数据库在设计和开发过程的优化 策略，主要包括基于数据稳定性的复制方案，基于高可用的负载均衡方案，基于 数据完整性的备份和恢复方案。同时总结了出现问题时的解决方案。

第五章:服务监控与应急措施优化。本章主要介绍了服务器层面的优化策略，主要包括多服务器的心跳监听服务配置，服务性能的监控脚本实现，基于 API 的 应急措施处置以及服务器安全性配置等方面的优化策略。同时总结了多种方案整 合使用的策略。

第六章:总结与展望。本章主要总结了不同优化策略对于系统性能的影响以 及在研究过程中获得的经验，并对优化策略的进一步研究做了简单的展望。