

**2025—2026学年第1学期**

**《专业工程实训4-3》阶段1**

**\*\*\*\*系统**

**优化提升报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 贡献度占比 | 学生得分 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**2025年11月5日**

**目录**

[1 数据库读写分离 1](#_Toc207821302)

[1.1 设计思路 1](#_Toc207821303)

[1.2 实现过程 1](#_Toc207821304)

[1.3 系统应用 1](#_Toc207821305)

[2 缓存机制 1](#_Toc207821306)

[2.1 设计思路 1](#_Toc207821307)

[2.2 实现过程 1](#_Toc207821308)

[2.3 系统应用 1](#_Toc207821309)

[3 远程方法调用Java RMI 2](#_Toc207821310)

[3.1 设计思路 2](#_Toc207821311)

[3.2 实现过程 2](#_Toc207821312)

[3.3 系统应用 2](#_Toc207821313)

[4 消息队列-RabbitMQ 2](#_Toc207821314)

[4.1 设计思路 2](#_Toc207821315)

[4.2实现过程 2](#_Toc207821316)

[4.3系统应用 2](#_Toc207821317)

[5 负载均衡 2](#_Toc207821318)

[5.1 设计思路 2](#_Toc207821319)

[5.2 实现过程 2](#_Toc207821320)

[5.3系统应用 3](#_Toc207821321)

[6 JAVA组件复用 3](#_Toc207821322)

[6.1 日志管理 3](#_Toc207821323)

[6.2 Excel文件导入导出 3](#_Toc207821324)

[6.3可视化展示 3](#_Toc207821325)

[6.4 自己扩展其他复用的组件 3](#_Toc207821326)

[7 容器化部署 3](#_Toc207821327)

[7.1 Docker安装 3](#_Toc207821328)

[7.2 微服务开发 3](#_Toc207821329)

1. 数据库读写分离

1.1 设计思路

介绍技术原理，读写分离的优势，可以如何应用到自己系统。

1.2 实现过程

建议使用Mysql数据库，至少两台机器，设计主从数据库，实现主从数据库同步。

主从数据库配置截图或配置代码，遇到的问题解决方案。

1.3 系统应用

如何在自己的系统中应用。

主要是针对读写操作如何实现的。

提供代码。

1. 缓存机制

2.1 设计思路

介绍缓存的技术原理，如何应用到自己系统。

2.2 实现过程

选择一个缓存服务器产品，Memcached或Redis

安装操作步骤截图，遇到的问题解决方案。

2.3 系统应用

如何在自己的系统中应用。

如何识别出热点数据的。

如何实现对缓存中热点数据更新的。

如何实现优先访问缓存、再访问数据库的。

1. 远程方法调用Java RMI

3.1 设计思路

介绍分布式对象的原理。

3.2 实现过程

两台机器实现远程访问。

3.3 系统应用

将自己系统中某些较为独立的处理封装为远程服务。

给出实现代码。

1. 消息队列-RabbitMQ

4.1 设计思路

介绍异步消息原理，消息队列服务器原理，自己系统的哪些操作可以应用消息队列服务器。

4.2实现过程

选择一个消息队列服务器，RabbitMQ或RocketMQ等。

安装操作步骤截图，遇到的问题解决方案。

4.3系统应用

如何在自己的系统中应用。

给出实现代码。

1. 负载均衡

5.1 设计思路

介绍负载均衡的技术原理，自己系统应用负载均衡的必要性。

5.2 实现过程

选用Niginx , 选用Tomcat等web服务器。实现对多个web服务器的负载均衡。

5.3系统应用

给出自己系统应用负载均衡的策略与配置代码

1. JAVA组件复用

6.1 日志管理

复用Log4j，介绍如何应用到自己系统。

6.2 Excel文件导入导出

介绍如何应用到自己系统。

6.3可视化展示

设计可视化大屏展示系统汇总数据。

6.4 自己扩展其他复用的组件

\*\*\*。

1. 容器化部署

7.1 Docker安装

介绍Docker的安装、配置过程步骤。

7.2 微服务开发

将已有业务操作抽取为微服务，并在Docker中部署。