

孟家伊

电话: 18973582105

邮箱: 1651665595@qq.com

求职意向: Java 后端开发



教育背景

华东师范大学 (985)

软件工程专业-本科生

2021.9 - 2025.6

技能特长

编程语言:

- **Java 基础:** 熟悉 JAVA 编程语言, 使用两年, 掌握集合框架、I/O 流、异常、反射等机制
- **Java 并发:** 熟悉 JAVA 并发容器和锁, 如 CAS、AQS 等, 掌握多线程、线程池、Thread Local 等原理
- **JVM:** 熟悉 JAVA 虚拟机, 掌握内存结构、垃圾回收机制, 类加载机制等, 了解过 JVM 调优方法
- **Web 框架:** 熟练使用 Spring、SpringMVC、SpringBoot、Mybatis 等主流框架, 熟悉 Spring AOP、IOC 等原理

数据库:

- **MySQL:** 熟悉 MySQL 基本使用和原理, 掌握索引、存储引擎、事务、MVCC、日志、锁等机制, 有 SQL 调优的经验
- **Redis:** 熟悉 Redis 的数据结构、线程模型、内存淘汰策略、持久化、分布式锁等机制, 熟悉 Redis 高并发缓存问题

基础知识:

- **数据结构与算法:** 掌握链表、栈、队列、树、图等数据结构, 掌握回溯、贪心、动态规划等算法。
- **计算机网络:** 熟悉 TCP、UDP、HTTP、HTTPS 等网络协议, 掌握 TCP 三、四次握手、流量控制、拥塞控制等机制。
- **操作系统:** 熟悉进程、线程、虚拟内存、I/O 多路复用等, 掌握进程间通信和多线程同步技术。

工具:

- **Linux:** 了解 Linux 常用命令, 如 grep、ps、top、free、df 等命令。
- **Git:** 熟悉 Git 多人协作开发及基本命令, 如 pull、push、merge、rebase 等

项目经历

AsyncCat 异步猫

2023.05 - 2023.08

项目介绍:

在实验室的音视频项目中, 发现审核、获取元信息、转码等一系列异步流程管理起来很麻烦, 在调研了业界的异步任务调度框架如 xxx-JOB、Activiti, 衡量了接入成本、业务需求以及开发难度, 决定抽象出 AsyncCat 异步猫框架用于调度。AsyncCat 是基于 java 开发的异步任务处理框架, 抽象了异步任务中任务管理、自动重试、优先级等非业务能力, 使异步任务开发成本低且高效。

个人职责:

- **架构设计:** 整体的框架搭建分为 **flowsvr 层** 和 **worker 层**, flowsvr 层提供 HTTP 服务, 任务创建、任务拉取、设置任务状态等接口, worker 层负责调度、消费任务
- **需求调研:** 调研了其他业务场景与其他异步处理框架, 发现无法满足需求, 选择抽象出异步猫这个框架。
- **数据库设计:** 设计低耦合的三张数据库表: 任务信息表、任务配置表、任务位置表, 方便任务快速注册、检索和管理
- **分表逻辑设计:** 根据异步任务框架的特点, 设计了一种全新的分表逻辑, **按照大小分表, 将冷热任务分开**
- **性能调优:** 对项目进行压力测试, 找到性能瓶颈并解决, 性能从一开始的 **500QPS 提升到 2000QPS**
- **架构优化:** 采用 **Redis 分布式锁** 替换 MySQL 本身的行级锁, 提高并发处理能力

技术难点:

- **分表:** 现有的分表组件大多采用分片设计。而在异步猫框架是 **需要按照大小来迭代新表的**, 并且 **老表逐渐作废**而不是多个分片长期提供服务, 相当于一个窗口不断的滑动。
- **性能调优:** 对接口进行压测后发现只有 500QPS, 通过 **MySQL 连接池、Redis 缓存、减少竞争**等手段优化至 2000QPS。
- **任务排序规则设置:** 框架抽象出一个 **order_time** 排序字段对任务排序, order_time 受到任务修改时间、任务优先级和任务失败次数等字段影响, 解决了排序规则和多个字段耦合的问题服务间调用
- **多机竞争:** 多个 worker 容易拉取到同一批任务, 使用 MySQL 行级锁解决时出现 CPU 不稳定的问题, 最终在 worker 侧引入 **Redis 分布式锁**。

兴趣伙伴

2023.09 - 2023.12

项目介绍:

兴趣伙伴是一个帮助大家有效地寻找合适的学习伙伴的 Spring Boot+Redis 移动端网站, 包括用户登录注册、按标签检索

用户、推荐相似用户、用户组队、房间公开加密等功能，解决用户在兴趣和需求组队上的问题，促进队友交流。。

个人职责：

- **用户登录注册**: 使用 Redis 实现分布式 Session，解决集群间登录态同步问题，采用阿里云 oss 进行上传用户头像
- **权限管理**: 利用 AOP 和自定义注解设计接口权限，通过全局异常处理器和统一返回类型、错误码，提高代码重用率
- **并发控制**: 为解决同一用户重复加入队伍、入队人数超限的问题，通过 **Redisson 分布式锁**解决并发问题，保证了接口幂等性
- **缓存优化**: 使用 Redis 缓存用户信息列表，将接口响应时长从 **1 秒缩短至 25 毫秒**，并通过自定义 Redis 序列化器保障数据完整性
- **代码简化**: 对项目中复杂集合处理(为队伍列表关联已加入队伍的用户)，使用 Java8Stream API和 Lambda 表达式简化编码
- **用户匹配**: 基于**编辑距离算法**实现了根据标签匹配最相似用户的功能，并通过优先队列来减少 TOP N 运算过程中的内存占用

技术难点：

- **性能提升**: 使用 EasyExcel 读取基础用户信息，利用**自定义线程池 + CompletableFuture** 并发编程，提高数据库批量导入性能
- **多机竞争**: 尝试多种方式解决多机竞争问题，如使用 MSQL 乐观锁，但会出现死锁问题，后采用分布式锁实现

个人评价

- 性格阳光开朗，工作积极认真、具有良好的团队合作精神和抗压能力强，曾参与数模比赛 30 小时持续奋战。
- 熟悉计算机语言特性，学习能力强，对于新语言能够快速上手，阅读官方文档后能够快速应用到项目中。
- 有较强的信息检索、问题解决能力，能够利用 GitHub Issues 区、ChatGPT、搜索引擎、博客等自主解决问题