

方衍林

✉ 邮箱: fangyanlin0525@outlook.com

☎ 微信: fyl9260

♂ 性别: 男

☎ 电话: 18938789260

📄 求职意向: GoLang后端开发

🎓 教育经历

南洋理工大学 (QS:12) 计算机控制及其自动化- 硕士 2025.01 - 2026.02

- 证书: IELTS

中南大学 (985) 交通设备与控制工程 - 本科 2019.09 - 2023.06

- GPA: 4.15/5.0 奖学金: 2021校三等奖学金
- 证书: CET-4、CET-6、优秀团员、志愿者证书

💻 专业技能

编程语言:

- GoLang: 熟悉slice、map、channel、interface等底层实现, 熟悉GMP模型、GC垃圾回收等机制。

后端组件:

- MySQL: 熟悉MySQL基本使用和原理, 掌握索引、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、持久化、过期键删除与内存淘汰策略、分布式锁等机制, 熟悉Redis高并发缓存问题的解决方案。

基础知识:

- 数据结构与算法: 掌握链表与数组、栈与队列、二叉树等数据结构, 掌握排序、回溯等算法。
- 计算机网络: 熟悉TCP/IP网络模型, 熟悉TCP、UDP、HTTP、HTTPS等网络协议, 掌握TCP三次握手、四次挥手、流量控制、拥塞控制等机制。
- 操作系统: 熟悉进程、线程、协程、虚拟内存、零拷贝、I/O多路复用等, 掌握进程间通信和多线程同步技术。

🏢 工作经历

中国中车株洲电力机车有限公司 后端开发工程师 2023.07 - 2024.03

工作背景:

- 在车体事业部信息化组负责中台服务及系统后端开发工作, 参与定时服务平台重构, 负责内部年度表彰大会(3000+人)的线下抽奖活动的落地实现。

个人职责:

- 主要负责Task Timer分布式定时服务平台项目重构, 负责主要模块的代码重构和存储结构的重构。
- 负责扩展分布式定时服务平台的功能以及平台在事业部的调试与维护。
- 负责设计并实现抽奖系统后端核心逻辑模块。

工作成果:

- 实现了Task Timer分布式定时服务平台, 并且平台的精准度误差控制在秒级, 支持2200QPS左右高负载服务。
- 扩展了定时服务平台的功能, 包括查询任务执行状态、任务重试机制。
- 通过定时服务平台为事业部提供工装全生命周期管理、质量数据定时上报、工艺参数定时抽检、定期检查工装设备状态的定时提醒服务。
- 通过抽奖系统为事业部年度表彰大会提供了稳定可靠、公平的抽奖服务。

📁 项目经历

TaskTimer分布式定时服务平台

项目介绍：事业部现有的定时服务与系统耦合度较高，为了解决事业部增加的定时需求，领导决定重构这个定时服务平台。Task Timer分布式定时服务平台是基于Go语言开发的一个定时系统，支持高精度，高负载的定时需求处理。

个人职责：

- 负责微服务整体设计：主要运用了Kratos框架等Go语言通用技术。
- 负责存储设计：通过运用MySQL+Redis的二级存储结构，数据分片，数据有序性，数据冷热分区等设计，有效解决了高精度，高负载等问题。
- 负责落地实现：通过引入模块化+异步化设计思路，让代码实现层次分明，模块职责清晰。有效提升了后续的扩展性和可维护性。
- 进行性能调优：为程序进行压力测试，找性能问题并且解决，性能从一开始的500QPS提升到2200QPS。

项目难点：

- 高精度：通过数据的分治，有序性，二级存储结构设计，为高频扫描提供基础，同时结合Go协程池等技术对任务处理速度进行提升，将定时任务的精准度控制在最大误差1秒左右。
- 高负载：指同一时间大批量任务触发。通过引入Redis ZSET对数据进行分片，为任务并发处理提供基础能力，再结合Go协程池等技术，使得高负载情况下任务也能快速处理。
- 数据冷热分区：有效地协调了冷热数据。通过定时迁移模块migrator的设计，让动态的热数据一直维护在缓存，方便进行高效存取，通过把冷数据放在磁盘或延迟冷数据的生成的设计也减少了对关键缓存资源的占用。

LuckyLottery抽奖系统

项目介绍：基于表彰大会抽奖系统的实践以及自身对其他业界方案的研究，扩展开发了这个系统。系统支持灵活配置不同奖品、概率和规则，适用于多种场景。

个人职责：

- 负责抽奖系统整体的架构设计，采用kratos框架实现抽奖系统，代码实现层次分明，模块职责清晰。
- 负责存储设计，通过运用MySQL+Redis的二级存储结构，完成抽奖逻辑中的中间数据和结果数据存储，有效解决了高性能，高负载等问题。

项目难点：

- 公平性策略：设置中奖概率区间，利用随机数匹配实现不同奖品中奖概率的差异性和不同用户中奖概率的公平性。
- 性能优化：采用不同的Redis数据结构对奖品、黑名单以及抽奖次数的优化，提升抽奖准入校验，奖品匹配发放逻辑的性能，性能由900QPS优化到3500QPS。
- 奖品池设计：基于Redis设计实现奖品池，配合发奖计划和定时奖品池填充任务来实现中奖概率的均匀分布。
- 并发控制：基于Redis设计实现分布式锁，用来保障抽奖接口的幂等性，防止同一个用户同一时间多次抽奖。

📝 个人评价

- 有快速学习和解决复杂问题的能力，善于利用CSDN、GitHub、Google等工具自主探索，思考并自主解决问题。
- 良好的沟通协作能力，在部门中能够进行有效的团队合作和技术沟通。
- 工作积极主动、认真负责、注重细节，具备较强的自驱力与执行力，能高效规划并推进工作目标。