

韦宇

性别：男

电话：18933549212

邮箱：3658043236@qq.com

CSDN: <https://blog.csdn.net/wy990880?type=blog>

年龄：21

学历：本科

求职意向：Golang开发



教育经历

广东药科大学

计算机科学与技术

本科

2022-09 ~ 2026-06

专业技能

- **Java基础**：熟悉Java编程语言、有两年使用经验、掌握**集合框架**、异常、多线程、反射等核心机制
- **Golang基础**：熟悉Golang语言、有一年使用经验、掌握Map、Channel、Select实现原理熟悉Gin、GORM组件
- **JVM**：熟悉JVM、掌握内存结构、**垃圾回收机制**，类加载机制，GC算法等，了解过JVM调优方法
- **MySQL**：熟悉MySQL基础原理、**存储引擎**、**索引原理**、MVCC、事务等机制、具备一定的SQL性能调优能力
- **Redis**：熟悉Redis底层数据结构、分布式锁、线程模型、**内存淘汰策略**等机制，熟悉**缓存击穿**、**穿透**、**雪崩**概念
- **计算机网络**：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
- **AI**：了解**AI Agent**、RAG、FunctionCall、LLM（如阿里百炼、DeepSeek）**Promot**管理和编排的基本概念及原理
- **AI Conding工具**：熟练运用如Cursor、通义灵码、**ChatGPT**、Claude等AI开发大模型工具

实习经历

美的集团

Java开发实习生

2024-11 ~ 2025-04

实习描述：参与开发数据供应链部门**APS系统（高级计划与排程系统）**，与美的集团**iPass（集成化生产管理平台）**、以及**MES（制造执行系统）**深度集成，实现**供应链计划→生产排程→车间执行**全链路数据协同

工作职责：主要参与项目核心链路开发、负责工单下达、工艺路线管理、车间排产的链路模块开发、完成核心代码编写工作

工作成果：

- **性能优化与缓存策略**：针对数据接口高频查询的性能瓶颈，采用**Caffeine**实现本地缓存，避免重复查询数据库，显著降低接口响应时间，将平均响应时间**从800ms减少到180ms**
- **多线程异步任务开发**：基于 **CompletableFuture**结合分页机制（**PageSize=1000**）实现数据拉取的并发处理，实现了**单批次1200条数据平均处理时间从7s优化至1.8s**
- **优化慢SQL**：通过**慢查询日志**和 **EXPLAIN** 分析，针对生产周期的执行依赖于任务的优先级排序，建立时间和任务优先级的**联合索引**，消除了file sort的影响，解决了需要频繁进行SQL查询的性能问题，并结合代码分批查询消除多表连接的影响，**查询时间从秒级优化到毫秒级**

项目经验

AsyncScheduler（异步调度框架）

项目背景：学校实验室的医学AI训练场景需要对收集到的图像进行数据采集、清洗、特征提取、分布式存储等多个步骤。为了提高开发效率，我抽象为多个异步任务并开发了一个基于Java的多阶段异步任务框架。

- 个人职责**：
- **负责架构设计**：采用**生产者-消费者模式**。整体框架分为**Flow Server（服务层）**和 **Worker（执行层）**。Flow Server向客户端（执行主要业务）和Worker层提供HTTP服务。主要接口有创建任务、拉取任务、轮询任务状态等。Worker负责消费任务
 - **任务调度设计**：支持**按动态优先级来调度任务**。综合创建时间、业务优先级、重试间隔（**采用渐进式间隔重试策略**）进行相对优先级排序
 - **数据库表设计**：**设计主要的三张数据库表**：任务信息表（存储具体任务实例并采用水平分表设计，以任务类型及ID+1命名）配置表（存储任务类型的通用调度参数）、位置表（通过读写指针来实现自动化分表）。方便任务快速注册和进行任务管理来**实现低耦合**
 - **任务治理设计**：定期扫描**状态**（如等待中、执行中、失败）的卡死任务将其进行状态恢复，提升容错能力
 - **架构优化设计**：多机竞争**由Mysql行级锁优化为Redis分布式锁**，下阶段考虑引入MQ，将任务拉取和执行解耦交给MQ

技术难点：

- **多机竞争方案及优化问题**：多个Worker去拉取任务容易拉到同一批任务。一开始这里在Worker侧引入Redis分布式锁来解决，任务冲突率解决90%，但Worker拉取和执行任务偶尔可能会CPU飙高至80%左右的问题。考虑**引入MQ**进行水平扩展效果

AI小智（Go+AI开发的智能知识库）

项目背景：作为技术社区群的活跃成员，我需要一个能自动总结日常技术讨论内容并构建可检索知识库的工具。于是开发了基于Go语言的AI智能助手、实现了知识库问答、数据分析、文档生成等功能。

- 个人职责**：
- **架构设计**：采用**Langchain**框架结合Go语言、设计了基于**RouterChain**的模块化架构。利用Go的并发特性实现系统分层（基础服务层、中间件层和应用层）确保各功能模块低耦合高内聚
 - **核心模块开发**：开发了AI对话记忆模块、实现上下文感知能力。通过Go语言的接口机制设计并实现了基于**outputparser**的响应解析流程、提高了系统对技术讨论内容的结构化处理能力
 - **知识库构建**：实现基于**MongoDB**作为向量数据库的**RAG检索系统**、通过worker池模式分配**embedding**任务，及协程数量动态调整、支持对技术讨论内容进行高效语义化存储和检索、使历史技术讨论可被精准获取

技术难点：

- **多模型兼容**：利用Go语言的接口抽象能力、设计model接口层实现了对**ChatGPT**、**阿里百炼**等多种大模型的统一调用
- **向量检索优化**：基于**MongoDB**向量数据库实现了高效的相似度检索引擎，采用余弦相似度算法进行语义匹配。通过Go的**channel**和**select**机制实现了并发检索请求的超时控制