

符尹铭人

手机: 13016064094 | 邮箱: fymingrengscut@163.com | Github | 个人网站 | 领英

教育背景

圣克拉拉大学

计算机科学与工程 硕士 GPA: 3.85

2024 年 9 月 - 预计 2026 年 6 月

圣何塞, 加利福尼亚州

华南理工大学

信息与计算科学 学士 GPA: 3.43 排名: 19/63

2020 年 9 月 - 2024 年 6 月

广州, 广东

技术技能

编程语言: C/C++, C#, Java, Python, Golang, JavaScript, HTML, SQL, Shell/Bash, CUDA

平台 & 框架: Spring Boot, Zookeeper, Kafka, gRPC, Muduo, Nginx, Netty, .Net Framework, WPF

数据库: MySQL, SQLServer, Redis, PostgreSQL, MongoDB

工具: Git, Docker, Kubernetes, Android Studio, AWS, gdb, cmake, TortoiseGit, Mercurial, Github Actions

工作经历

C++ 开发工程师

2023 年 6 月 - 2023 年 10 月

中望软件

广州, 广东

- 基于 **STL** 和 **Boost** 设计并实现了一个高性能、低延迟的内存池, 适用于 3D 工程设计软件。
- 采用**单例模式**与**简单隔离存储策略**, 为顶点、线段和多边形对象提供**自定义的 malloc/free 和 new/delete** 操作, 将 3D 模型的生成与导出速度提升 **40%** 以上。
- 利用**完美转发**与**模板**消除不必要的拷贝开销, 使用 **std::atomic** 实现无锁指针操作, 并结合 **std::mutex** 确保内存分配与回收的线程安全性。
- 通过**内存对齐**减少内部碎片, 采用**连续块预分配**降低外部碎片, 并使用 **placement new** 在预分配块上复用对象, 使 3D 几何数据的内存占用降低 **15%**。

项目经历

基于 Raft 共识算法的分布式 KV 数据库

2024 年 12 月 - 至今

框架: C++, Boost, STL, Muduo, protobuf

- 基于 **Raft 共识算法**实现了**日志复制**与**领导者选举机制**, 利用**线程池**管理心跳检测与选举任务, 确保日志一致性与集群稳定性。
- 设计并开发了基于 **Protobuf** 的 **RPC 框架**, 实现 Raft 节点间的远程过程调用与高效序列化数据传输。
- 构建基于**跳表**的键值数据库, 提供高性能的数据存储与检索能力。

MathMind: 基于 LLM 的数学助手-扫描 & 即时求解

2023 年 12 月 - 2024 年 5 月

框架: Python, Java, Kotlin, SQL, Flask, PyTorch, Android Studio, Docker

- 设计并实现基于 LLM 的可扩展数学解题**微服务**, 后端采用 **Python/PyTorch/Flask** 技术栈, 移动端使用 **Java/Kotlin** 开发。
- 开发 **RESTful API** 和 **WebSocket** 连接。使用 **OpenCV** 和 **Pickle** 解码并预处理 **Base64** 编码的图像, 并集成基于 Transformer 的 **LaTeX OCR 模型**, 用于扫描服务中的文本和公式识别。
- 使用**监督式微调 (SFT)** 在代数和微积分数据集上对 ToRA-7B 和 Llama3 进行了微调, 并通过 **TensorRT-LLM** 部署本地模型, 并集成远程 GPT 和 Gemini API, 共同构建求解服务。
- 构建 **Docker** 镜像, 使用 **Miniconda** 管理多版本 Python 环境以进行子进程调用, 通过 **Docker Compose** 协调部署, 并使用 Shell/Bash 脚本自动化部署任务。
- 基于 **MVVM 架构**开发 Android 应用, 利用 **Kotlin Coroutines** 处理异步任务, 使用 **Handlers** 更新 UI, 并通过 **SQLite** 存储历史解题记录, 实现本地数据持久化。

基于 Muduo 的高并发 C++ 服务器库

2023 年 4 月 - 2023 年 7 月

框架: C++, Muduo, Boost, STL

- 采用**非阻塞 I/O**、**多路复用**及 **Reactor 模式**, 基于 **Muduo** 和 **Boost** 设计并实现了高并发服务器。
- 开发 **EventLoop**、**Poller** 和 **Channel** 等核心组件, 实现**事件驱动循环监听**、**请求分发**与**异步事件处理**。