何清

17373546038 heqing2023@email.szu.edu.cn hq876677504

求职意向：C++后端开发工程师

教育经历

深圳大学 硕士 计算机技术 2023.09-2026.07

主修课程：机器学习、算法设计与分析、系统软件原理与实践、无线网络技术、计算机前沿技术、脑与认知科学。

专业技能

- 熟悉Linu系统编程，如进程管理（fork/exec）、IPC机制、POSIX线程库等，具备多线程同步开发能力；
- 熟悉Linux网络编程，I/O多路复用机制，能够基于Reactor/Proactor模式构建高并发网络服务；
- 熟悉OSI七层网络模型、TCP/IP协议栈等；
- 熟悉数据结构及常见算法，如数组、链表、栈、队列、二叉树、查找与排序、BFS和DFS等；
- 熟悉MySQL基本操作，表的约束、范式、索引，具备数据库设计能力；
- 熟悉MVVM软件架构模式，具备使用Cpp和QT5实现业务逻辑与UI层高效解耦的应用开发能力；
- 了解xv6操作系统源码，通过修改内核代码实现进程调度、IPC和信号量机制，实验均记录在Blog中；
- 熟悉Git等开发工具，软件设计师（中级）。

实习经历

深圳市智物联网络有限公司 后端开发工程师 2025.07-2025.09

- **负责API分组与信息管理模块的开发测试：**实现API分组管理和API信息管理，优化API分组管理的多叉树生成算法，处理父子节点关系；使用Apipost完成接口测试，为Yunuis平台提供API层级管理能力；
- **研究与实现微服务架构下的HTTP请求转发：**基于开源工具Consul实现微服务的注册与发现，获取所有微服务的Action列表到Yunuis网关层常驻内存，实现精准路由至对应微服务执行业务逻辑；
- **设计并实现多类型Action的请求分发与执行机制：**基于工作一和二，通过哈希表缓存内部、系统、用户自定义的Action列表；内部Action利用Go反射机制，查找并执行对应结构体方法；系统Action复用已开发的微服务HTTP转发器进行路由；用户自定义Action调用Mixgo框架的Lua脚本引擎执行用户自定义逻辑。

项目经历

分布式实时消息推送系统 后端开发 2024.10-2025.09

项目描述：分布式实时消息推送系统通过三级解耦架构，实现Comet层，Job层和Logic层，使用Kafka和gRPC实现服务间通信；将连接管理、业务逻辑与消息推送分离，为需要高并发、低延迟消息推送的场景提供稳定可靠的后端服务；本系统支持房间隔离消息、全局广播等多种消息模式，并提供完整的用户注册、登录认证、消息持久化与历史消息查询等功能。

部署上线：https://www.utopianyouth.top

源代码：https://github.com/UtopianYouth/myself_chatroom_distribute

Comet层：高效管理WebSocket长连接，并将消息实时、可靠地推送到客户端

- **基于muduo网络库的Reactor模型：**通过非阻塞I/O和Reactor事件驱动模式，高效处理并发连接；
- **连接与会话管理：**通过ConnectionManager管理所有客户端WebSocket连接，使用PubSubService类管理房间与用户的订阅关系，实现高效地广播消息推送；
- **gRPC服务端：**将gRPC接口给Job层调用，解析Job层的Protobuf消息，通过订阅管理机制找到对应连接进行消息推送。

Job层：从Kafka获取推送消息，调用Comet层消息推送逻辑，实现业务逻辑与消息推送解耦

- **Kafka消费者：**使用librdkafka高效从Kafka指定Topic中消费消息，保证消息顺序性和可靠性；
- **消息路由与转换：**解析Protobuf消息(PushMsg)，根据type和roomId字段，路由到正确的推送逻辑；
- **gRPC客户端：**维护与Comet层的gRPC连接，根据消息类型，调用不同的gRPC方法，将消息分发到一个或多个Comet服务器上。

Logic层：处理所有HTTP请求、实现业务逻辑和操作存储层，如用户注册登录、消息存储、房间管理等

- **数据持久化：**使用MySQL进行数据持久化存储，Redis缓存session和热点数据；
- **Kafka生产者：**消息广播业务逻辑完成，构造Protobuf格式消息，通过librdkafka异步推送到Kafka中，通知Job层进行推送。

基于Reactor网络模型的KV存储系统

后端开发　　2024.07-2025.08

项目描述：该项目是基于Reactor网络模型和线程池架构开发的高性能KV存储演示系统，采用前后端分离设计，实现了支持Web界面交互的键值存储服务；后端使用C++实现纯API服务器和三KV存储引擎，前端通过Nginx托管静态资源并反向代理API请求，最后，测试其性能达到10~15kQPS。

部署上线：https://www.utopianyouth.top/kv/

源代码：https://github.com/UtopianYouth/kv-webserver

- **Reactor网络模型**：主线程使用epoll边缘触发模式监听客户端连接和数据事件，工作线程池异步处理HTTP请求解析和KV命令执行，实现10K+高并发网络通信；
- **前后端分离架构**：Nginx托管前端静态资源并反向代理API请求到C++后端，后端专注API服务和业务逻辑；
- **HTTP协议解析**：采用主从状态机配合解析HTTP协议，支持GET/POST方法和JSON数据交互；
- **多引擎KV存储**：实现Array、Hash、RBTree三种存储引擎，支持SET/GET/DEL/MOD/EXIST命令，满足不同性能需求场景；
- **细粒度锁优化**：为KV引擎配备独立std::shared_mutex，读操作使用共享锁支持并发执行，写操作使用独占锁保证数据一致性；
- **零拷贝技术**：使用mmap内存映射和writev分散写技术，减少数据在用户态和内核态之间的拷贝次数，提升静态文件响应性能。

多任务RHEDD图像分析方法研究

客户端开发&图像处理　　2024.07-至今

项目描述：该项目是硕士毕业论文研究课题，属于正在进行中项目，目前使用了 OpenCV4.8 传统图像处理方法完成了图像对象检测任务，使用QT5完成UI设计。

源代码：https://github.com/UtopianYouth/RHEEDAnalysis

- 使用OpenCV图像融合、锐化、局部阈值二值化等提取图像中的亮斑（对象），标定亮斑在图像中的位置；
- 使用QT5常见UI组件、多线程等完成UI界面设计；
- 使用Model View View Model(MVVM) 软件架构模式，实现算法层和UI层的解耦；
- 使用工厂模式对多种算法进行更上层的抽象（多态），实现多种算法的集成。