

黄晓强

✉ 2939349707@qq.com

☎ 18815978979

👤 C++/Go

教育经历

福州大学 2021.09 ~ 2025.06
计算机与大数据学院 计算机科学与技术

实习经历

腾讯科技（深圳）有限公司 2024.06 ~ 2024.09
TEG研发管理部 后端开发实习生

应用技术：Golang，TRPC-go，MySQL，Redis，Git
实习业务：参与研效平台的CICD核心子系统——交付流项目的深度开发与优化，致力于提升腾讯内部各BG的研发效能与系统稳定性
主要工作：设计并实现执行系统的限流机制：采用预分配与失败补偿的方式更新并发额度以保证严格限流，通过Redis原子计数与Lua脚本完成并发额度的原子性变更，同时基于ZSet实现延时调度模块保障实例任务的正确调度；
对交付流项目做可灰度改造：通过分析交付流子系统的流量调度，开发网关插件、改造定时任务、编写灰度模块，优化redis灰度名单查询逻辑，实现系统按租户进行灰度的能力，保障交付流的稳定性和正确性；
参与MySQL慢查询排查：通过修改查询逻辑和索引解决深分页问题带来的性能损耗，显著提升查询效率，将总查询时间由3min缩短至0.01sec

专业技能

GitHub：<https://github.com/kaoyu12138> Blog：pinoky - 博客园 (cnblogs.com)

编程语言：熟悉C/C++，熟悉STL常用容器及其底层原理，C++11常用特性(智能指针等)，熟练使用C++的指针应用与内存管理，C++的封装继承与多态
熟悉Golang，掌握Go语言SOLID设计理念，了解map，slice，channel等实现原理，了解GMP模型调度器，gc垃圾回收，内存逃逸等底层原理
操作系统：熟悉操作系统的进程通信、线程同步、虚拟内存实现等内容，熟悉无锁化编程以及wait free思想，熟悉协程实现原理，了解常见内存池TCMalloc，熟悉Linux下的内存布局与内存管理，了解Linux常用命令与性能排查手段
通信技术：熟悉IO多路复用技术(select/poll/epoll)底层原理，熟悉Linux五种网络IO模型，熟悉Recator网络IO框架，了解io_uring及DPDK等高性能IO技术
网络协议：熟悉TCP/IP，UDP，HTTP/HTTPS，QUIC等网络协议，熟悉TCP三次握手，四次挥手，流量控制，拥塞控制等手段
分布式理论：熟悉共识算法Raft，了解分布式CAP理论以及分布式系统设计原理
中间件技术：熟悉高性能RPC框架设计原理(协议封装、服务发现、负载均衡、熔断限流等)，了解protobuf，ZooKeeper等组件的基本使用与底层实现
数据库技术：熟悉MySQL中InnoDB索引使用与底层原理，熟悉MySQL的日志系统(redolog、binlog、WAL技术)，MVCC版本控制，掌握MySQL相关性能调优方法，熟悉事务ACID特性及高可用架构(主备一致、读写分离)
熟悉Redis数据类型使用场景与底层原理、熟悉主从库与切片集群机制、熟悉缓存异常处理(缓存穿透、缓存击穿、缓存雪崩、缓存不一致等)

项目经历

分布式网络通信RPC框架

应用技术：C++，Protobuf，muduo，ZooKeeper，CMake
项目难点：服务注册与服务发现，数据序列化处理，TCP沾包问题，高性能实现
项目描述：该RPC网络通信框架用于实现分布式场景下本地服务在RPC节点上的注册、发布与远程调用功能，主要模块分为服务注册，服务发布，服务调用以及日志的异步输出
主要工作：使用protobuf实现RPC方法的注册以及数据序列化处理，使用epoll多路复用与Reactor框架高效管理服务发布端与调用端的网络连接，通过ZooKeeper实现服务注册与服务发现，设计合理的RPC协议解决TCP沾包问题。通过LALB负载均衡算法完成对下游服务的自适应分流，通过令牌桶算法保证平滑限流，初步实现了RPC框架的服务治理功能
项目总结：学习了分布式网络通信RPC框架的设计原理，掌握protobuf、muduo、ZooKeeper的使用方法，深入学习主流RPC框架源码(grpc，trpc，brpc等)
GitHub：<https://github.com/kaoyu12138/mprpc>

分布式系统一致性算法Raft 以及 基于Raft 的KV服务(MIT 6.824)

应用技术：Go，RPC，Raft，MapReduce
项目难点：数据竞争与并发安全处理，多线程下Debug，共识算法实现
项目描述：Raft算法用于实现数据一致性并保障分布式系统中系统的容错能力，主要模块分为领导人选举，日志复制，日志持久化，日志压缩以及状态机实现的KV存储服务
主要工作：使用协程周期性检测选举超时和已提交的Log，通过RPC通知Follower更新自己的状态。实现了基于半数投票原则的领导人选举，以及针对投票逻辑的选举限制。实现了状态机并且会在Raft的状态达到设定时进行快照(日志压缩)，并基于复制状态机实现了KV存储服务
项目总结：学习了分布式系统中数据一致性的原理，领导人选举、日志复制、日志持久化、日志压缩等功能实现良好，积累了设计与调试分布式系统的经验
GitHub：<https://github.com/kaoyu12138/MIT6.824-lab2>
<https://github.com/kaoyu12138/MIT6.824-lab3>

个人总结

- 在开发中对待问题有认真求索的精神，能够在短时间内发现并解决问题。同时善于总结整理，在项目结束后会严格要求自己规范书写README文件，实习期间沉淀多篇技术文档
- 有较强的信息检索能力，能通过Google/Github等国内外论坛/文档解决技术问题
- 有较强的自我管理能力，自学能力较强，在学习或开发项目过程中能合理规划安排稳步推进