

姓名：王泽洲

🎓 教育经历：兰州理工大学 -- 计算机科学与技术 -- 2024.09_至今 -- 大二在读

📞 微信 / 电话：153-7544-4138 | 📧 邮箱：wang_ze_zhou@163.com

👤 年龄：19 | 📍 所在地：西安 / 兰州 | 🌟 英语等级：CET - 4

👤 求职意向：C / C++ 后端开发工程师 | 🌐 个人网站：wzz6423.com

🔗 gitee：gitee.com/wzz6423 github：github.com/wzz6423



wechat

个人网站



⚡ 专业技能

编程基础

C / C++

- | 熟悉 C / C++ 基础语法、关键字、指针、引用等，并了解一定底层原理
- | 熟悉类和对象，动态内存管理，封装、继承、多态、模板等泛型编程，并了解一定底层原理
- | 熟悉容器：string、vector、list、queue、priority_queue、stack、set、map、unordered_set、unordered_map、bitset
- | 熟悉 C++11、14、17、20 常用新特性、语法糖：auto关键字、右值引用与完美转发、lambda、函数包装器、智能指针与 RAII、线程库、锁、原子操作、文件系统库、文件操作
- | 熟悉常用设计思路、设计模式、Debug 方法、动静态库封装与使用、C / C++ 程序产生与运行原理、CMake与Makefile

Python

- | 熟悉 Python 基础语法、元组、列表、字典、类和对象、装饰器语法及脚本编写
- | 熟悉 Python 包管理及其工具 pip，熟悉 Python 部分常用第三方库的使用

数据结构 & 算法

- | 熟悉常用数据结构：顺序表、链表、二叉树、栈、队列、堆、AVL 树、红黑树、字典树、并查集、位图、B-树系列、跳表
- | 熟悉常用排序算法：直接插入排序、希尔排序、直接选择排序、堆排序、冒泡排序、快速排序不同版本及优化、归并排序（内、外排序）、计数排序、基数排序

操作系统 & Linux

- | 熟悉常用开发工具：常用指令使用、实用工具使用、shell 脚本编写
- | 熟悉文件系统：文件创建、权限管理、“一切皆文件”及部分原理、文件描述符与inode、软硬连接、动静态库、重定向
- | 熟悉进程相关：进程地址空间、进程O(1)调度算法、进程替换、进程间通信（管道、消息队列、共享内存、信号量）、信号控制及其原理、进程池、守护进程
- | 熟悉线程相关：轻量级进程、线程安全、同步互斥、pthread 库、mutex 互斥锁、cond 条件变量、线程池、读写锁、了解CAS 无锁编程
- | 了解操作系统其它相关内容：用户态、内核态、虚拟地址与物理地址、页表与 MMU、文件落盘、页框页帧、硬件中断、软中断、局部性原理

网络

- | 熟悉 OSI 七层协议、Udp、Tcp、协议序列化与反序列化、套接字、HTTP / HTTPS、签名与摘要、加密、证书
- | 熟悉 select、poll、epoll、reactor 多路转接（LT、ET 模式）、同步异步、长短连接
- | 了解 Ip 地址、Mac 帧、端口号、路由器、交换机、ARP 协议、NET 机制、内网穿透、代理服务器

中间件

数据库

- | 熟悉 MySQL 常用操作：数据库与表的增删查改（函数、多表查询、limit 分页、自连接、子表查询、表的内连和外连）
- | 熟悉 MySQL 索引、事务、用户管理与权限控制
- | 熟悉 Redis 常用操作：键值对、列表、哈希等增删查改操作

Git

- | 熟悉 Git 常用操作、版本控制与操作、分支管理、本地仓库与远程仓库的相关操作
- | 了解 Git 与开发常见分支策略

序列化与反序列化

- | 熟悉 ProtoBuf3 常用基础语法与编译命令及对编译生成源文件的使用
- | 熟悉 Json、YAML 编写与使用

Docker

- | 熟悉 Docker 镜像制作、上传、下载、迁移、使用等操作

其它框架

- | 熟悉日志模块(spdlog、logging)、消息队列(RabbitMQ)、RPC框架(brpc)、功能性云平台(短信、语音识别云平台)使用

测试

- | 熟悉白盒测试、黑盒测试、灰盒测试、手工测试、自动化测试(接口、网页)
- | 熟悉测试用例设计、测试工具（gflags、gtest、allpairs、selenium、requests、pytest、Json Schema、Jmeter、Apifox、allure）的使用、测试报告编写
- | 了解性能测试

项目经历

StellarPost

项目链接: <https://gitee.com/wzz6423/my-code/tree/master/project/StellarPost>



功能实现:

| 仿照主流聊天软件微信实现了一个分布式微服务通讯后端系统

核心技术:

| 设计实现入口网关、用户、好友、文件、消息存储、消息转发、语音识别分布式微服务模块
| 项目框架代码使用 protobuf 进行编写, 测试用例使用 gtest 框架, 配置文件使用 gflags 框架, 日志使用 spdlog 框架
| 分布式微服务模块之间的服务注册与服务发现使用 etcd 框架, 微服务间 rpc 通信使用 brpc 框架
| 短信验证码及会话管理使用 Redis, 消息存储微服务消息队列使用 RabbitMQ, 数据库使用 MySQL+ODB 进行 ORM
| 服务端与客户端间的常规通信使用 httplib 库, 服务端主动事务通知使用 websocket 框架
| 好友搜索、消息记录搜索使用 Json + elasticsearch 框架
| 短信验证码平台采用阿里云短信云平台, 语音识别平台采用百度语音识别技术云平台
| 项目构建采用 CMake, 项目部署使用 shell/Python 脚本 + docker(dockerfile + docker-compose.yml)
| 支持守护进程, 具有良好的功能扩展性及 C++23 新标准特性扩展

LoomAlloc

项目链接: <https://gitee.com/wzz6423/my-code/tree/master/project/LoomAlloc>



功能实现:

| 仿 tcmalloc 库实现的 mini-tcmalloc, 通过三级缓存架构、无锁设计、高效内存对齐和碎片控制算法实现了一个高性能、高并发的内存池, 能显著优化内存分配性能, 减少系统调用和锁竞争开销, 适用于高并发场景

核心技术:

| 三层缓存架构: ThreadCache (线程本地缓存)、CentralCache (中心缓存)、PageCache (页缓存)
| ThreadCache (线程本地缓存) 使用 TLS (线程本地存储) 实现了线程独立缓存、消除锁竞争, 使用慢启动反馈机制动态调整从 CentralCache 申请的内存块数量以平衡空间与时间
| CentralCache (中心缓存) 使用单例模式协调所有线程的内存操作, 使用 RAII 的哈希桶锁减小锁粒度在保证安全的同时提升性能, 按需从 PageCache 申请大块内存切分后分配给线程
| PageCache (页缓存) 使用单例模式, 小对象以页为单位申请分配内存, 合并相邻页减小外碎片, 使用基数树高效实现页号到内存地址的映射且无需锁, 直接调用系统内存申请接口
| 计算索引使用位运算, 内存块大小切分按照梯度对齐优化内碎片, 固定大小对象使用定长内存池避免频繁使用系统调用, 超过 256KB 大小内存直接通过 PageCache 向系统申请内存直接使用、避免多层缓存开销以提升性能
| 支持 Windows 和 Linux 跨平台, 支持 32 位与 64 位架构

NetServer

项目链接: <https://gitee.com/wzz6423/my-code/tree/master/project/NetServer>



功能实现:

| 仿 muduo 库实现的 mini-muduo, 构建了一个高性能、高并发的 HTTP (支持协议切换) 服务器, 能够高效的处理大量并发连接和请求, 适用于各种需要高性能网络服务的场景

核心技术:

| 支持 GET、POST、PUT、DELETE 多种 HTTP 协议常用请求方法并支持扩展, 支持静态资源与动态方法请求处理
| 使用线程池和 EventLoop 模块来管理多个事件循环线程, 每个线程负责处理一部分连接, 提高了服务器的并发处理能力
| 使用 epoll 实现高效的事件多路转接, 能够监控多个文件描述符的事件无需轮询, 提高了服务器的响应速度和吞吐量
| 通过非阻塞套接字和事件回调, 避免了因等待 I/O 操作而阻塞线程, 进一步提高了服务器的并发处理能力
| 使用时间轮来管理多个定时任务, 提供了高效的定时任务调度机制
| 实现了一个通用的 Any 类型容器, 可以存储任意类型的值, 用于 HTTP 请求的接收处理上下文
| 实现了 Buffer 动态的字节缓冲区, 支持读写操作, 简化了网络数据的处理过程

NexusLogger

项目链接: <https://gitee.com/wzz6423/my-code/tree/master/project/NexusLogger>



功能实现:

| 仿 spdlog 库实现了一个高性能的基于同步、异步, 日志数据支持不同落地方向的日志系统

核心技术:

| 支持不同日志级别的日志输出及默认日志输出级别的限定
| 提供三个默认日志落地方向 (标准输出、单一文件、按照文件大小/时间的滚动文件) 并支持自定义扩展日志落地方向
| 支持多线程并发写日志, 日志数据同步异步落地模式

CloudDisk

项目链接: <https://gitee.com/wzz6423/my-code/tree/master/project/CloudDisk>



功能实现:

| 实现了一个支持断点续传的 PC 端和 web 端的文件云备份系统

核心技术:

| 支持选择文件夹下的文件自动进行检测并备份至服务器
| 支持通过 web 端访问、下载备份的文件, 支持 web 端上传文件
| 支持 web 端断点续传

Searcher

项目链接: <https://gitee.com/wzz6423/my-code/tree/master/project/Searcher>



功能实现:

| 实现了一个根据正排索引与倒排索引的站内搜索引擎

核心技术:

| 根据正排索引与倒排索引构建的站内搜索引擎
| 支持通过 web 端访问, 使用 jieba 分词支持分词技术以及暂停词屏蔽