

高祥

手机 178-6519-7981
邮箱 gaoxiangnumber1@126.com
Github <https://github.com/gaoxiangnumber1>
博客 <http://blog.csdn.net/gaoxiangnumber1>
学历 山东大学—计算机科学与技术专业—本科

项目

网络库 netlib <https://github.com/gaoxiangnumber1/netlib>
Reactor 使用 non-blocking IO 配合 IO multiplexing 配合应用层 Buffer。用户注册接受连接、接收数据的 Event Callback, IO 线程运行 Event Loop, 以 Event Driven 的方式实现业务逻辑。
Multithread 支持 4 种 IO 计算模型, 默认使用 one loop per thread 配合 thread pool。主 IO 线程接受连接并分配至子 IO 线程, 连接的 IO 任务由所属的 IO 线程完成, 计算任务由 thread pool 完成。
EventLoop Epoller 使用 epoll level-trigger 模式实现 IO 复用, Channel 负责注册和分发事件回调, TimerQueue 使用 timerfd 实现 Add/CancelTimer。EventLoop 使用 eventfd 实现异步唤醒 IO 线程执行任务回调, 使用 RunInLoop 实现无锁、线程安全的跨线程调用。
Thread Lib 使用 RAII 手法封装非递归 mutex 做到 Scoped Locking。Condition 解决虚假唤醒, 区分 signal/broadcast 语义。Thread 支持创建/销毁线程, 使用 tid 标识线程。使用任务队列实现适合多消费者的 ThreadPool。
Buffer 使用动态 char 数组保存数据, 支持前方高效地添加数据、内部移动数据, 大小自动适应。
Read 使用 TcpConnection 的 input buffer、IO 线程的 stack buffer 配合 readv() 实现节约内存、效率高且公平的数据读取。
Write 使用 TcpConnection 的 output buffer 配合 EventLoop::RunInLoop 实现非阻塞、线程安全且保证消息完整的数据发送。
Connection 使用 std::shared_ptr 管理 TcpConnection 的生命期; 使用 shutdown(2) 关闭连接, 保证收发数据的完整性。
TcpClient 能主动发起 TCP 连接, 带 back-off 地重试直至建立连接; 能在连接断开后自动重新连接; 能主动断开连接。

技能

进程线程 熟悉实现原理、IPC 共享内存、线程同步方法, 熟悉多线程编程。
虚拟内存 熟悉地址空间与翻译, 页式内存管理, 内存映射, 动态内存管理。
TCP 协议 熟悉建立/关闭连接, 确认、超时/快速重传, 流量控制, 拥塞控制。
网络编程 熟悉 TCP socket 编程流程, 常见服务器并发模型, IO 复用与原理。
内存管理 熟悉智能指针, 类的构造与拷贝, 右值引用、移动语义, new/delete。
数据结构 熟悉字符串、动态数组、链表、栈、队列、二叉树、优先队列。