

Linux 轻量级高并发 Web 服务器

项目介绍

在 Linux 环境下使用 C++ 搭建轻量级高并发 Web 服务器，支持一定数量的客户端连接并及时响应

内容

使用**同步 IO 方法 + epoll + 线程池**的方式，模拟 **Proactor** 模式来实现服务器高并发处理

通过使用**有限状态机**，实现对 HTTP 请求的读取和解析，从而实现服务端和客户端之间的通信

利用**双向链表**管理**定时器**及时关闭不活跃的连接

使用 Webbench 对服务器进行**压力测试**，实现一万左右的并发连接

1. 项目如何实现
2. **select** 和 **epoll** 的区别，**select** 和 **epoll** 的原理
3. 同步和异步的区别，Linux下的IO模型
4. 为什么使用 **Proactor** 模式不使用 **Reactor** 模式，边缘触发和条件触发的区别
5. 进程和线程的创建方式
6. 为什么不使用进程池，线程池的优点
7. 进程间通信的方式，多个线程访问临界资源如何解决
8. 什么是僵尸进程
9. 有限状态机
10. 双向链表的操作，定时器
11. **tcp** 连接的建立和释放，**tcp** 和 **udp** 的区别
12. **http** 和 **https** 的区别，在浏览器中输入网址建立连接的过程
13. 数据库，**MySQL** 和 **Redis** 使用过吗