

# C/C++Linux服务器开发

## 高级架构师课程

三年课程沉淀

五次精益升级

十年行业积累

百个实战项目

十万内容受众

讲师: darren/326873713



助教: 秋香/ 3007537190



# 讲师介绍——专业来自专注和实力



Darren老师

曾供职于国内知名半导体公司（珠海扬智/深圳联发科），曾在某互联网公司担任音视频通话项目经理。主要从事音视频驱动、多媒体中间件、流媒体服务器的开发，开发过即时通讯+音视频通话的大型项目，在音视频、C/C++/GO Linux服务器领域有丰富的实战经验。

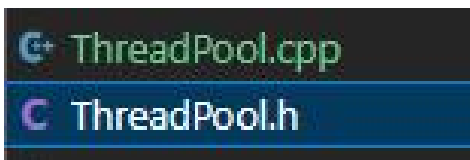
## 6-图床的线程池、文件数量计数...等代码优化

1. 添加线程池的支持
2. 多线程模型下数据收发处理
3. 使用redis优化文件数量计数

# 1. 添加线程池的支持

使用C++11线程池

具体见课上分析



```
83 int initHttpConn(uint32_t thread_num)
84 {
85     g_thread_pool.init(thread_num); // 初始化线程数量
86     g_thread_pool.start();          // 启动多线程
87     netlib_add_loop(http_loop_callback, NULL); // http_loop_callback被epoll1所在线程循环调用
88     return 0;
89 }
```

## 2 多线程模型下数据收发处理

在线程池处理任务时需要回发给客户端的数据加入到发送队列，由`epoll`所在线程取出然后回发给客户端。

具体见课上分析

```
else if (strcmp(url.c_str(), "/api/login", 10) == 0)
{
    // 登录
    auto f1 = g_thread_pool.exec(std::bind(&CHttpConn::HandleLoginRequest, this,
        std::placeholders::_1, std::placeholders::_2), url, content);
    // HandleLoginRequest(url, content);
}
```

通过线程池处理

```
398
399 int CHttpConn::HandleLoginRequest(string &url, string &post_data)
400 {
401     LOG_INFO << "into";
402     string str_json;
403     int ret = ApiUserLogin(url, post_data, str_json);
404     char *szContent = new char[HTTP_RESPONSE_HTML_MAX];
405     uint32_t nLen = str_json.length();
406     snprintf(szContent, HTTP_RESPONSE_HTML_MAX, HTTP_RESPONSE_HTML, nLen, str_json.c_str());
407     CHttpConn::AddResponseData(m_uuid, szContent);
408     delete szContent;
409
410     return 0;
411 }
```

将要回发客户端的数据入队

```
void CHttpConn::AddResponseData(uint32_t conn_uuid, string resp_data)
{
    LOG_INFO << "into";
    ResponsePdu_t* pResp = new ResponsePdu_t;
    pResp->conn_uuid = conn_uuid;
    pResp->resp_data = std::move(resp_data);

    s_resp_mutex.lock();
    s_response_pdu_list.push_back(pResp);
    s_resp_mutex.unlock();
}
```

```
void http_loop_callback(void* callback_data, uint8_t msg, uint32_t handle, void* pParam)
{
    UNUSED(callback_data);
    UNUSED(msg);
    UNUSED(handle);
    UNUSED(pParam);
    CHttpConn::SendResponseDataList(); // 静态函数，将要发送的数据循环发给客户端
}
```



## 3 使用redis优化文件数量计数

使用redis记录

- 我的文件数量
- 共享文件数量
- 分享图片数量

```
#define FILE_PUBLIC_COUNT "FILE_PUBLIC_COUNT" // 共享文件数量  
#define FILE_USER_COUNT "FILE_USER_COUNT" // 用户文件数量 FILE_USER_COUNT+username  
#define SHARE_PIC_COUNT "SHARE_PIC_COUNT" // 用户分享图片数量 SHARE_PIC_COUNT+username
```

要考虑的重点：

Redis没有记录怎么办，该如何处理。

这个大家先理解原理，在前几节课代码增加过程中，有些使用redis逻辑的没有写上，这里大家可以考虑做补充。