# 试用期工作内容

## 熟悉代码

熟悉了解接入层代码以及框架，了解日志的查看，为接下来对接入层服务器偶发错误的解决以及对接入层服务器的性能优化打下了坚实的基础。

## 服务器压测

协助导师进行了服务器数据的压测，测试了客户端在非安全或安全的状况下，一台服务器最大能接受多少连接数以及服务器跟特定个客户端建立连接的时间，通过改变服务器的参数，包括worker线程数，io线程数、mysql\_pool、redis\_pool等，使服务器能接收的客户端数最大、接收特定数目客户端的时间最短。在一定程度上优化了服务器性能。

## 编写客户端

用go编写了一个支持建立多个连接的客户端，以替代之前的python客户端，获得更好的效率，更方便了以后的服务器测试。

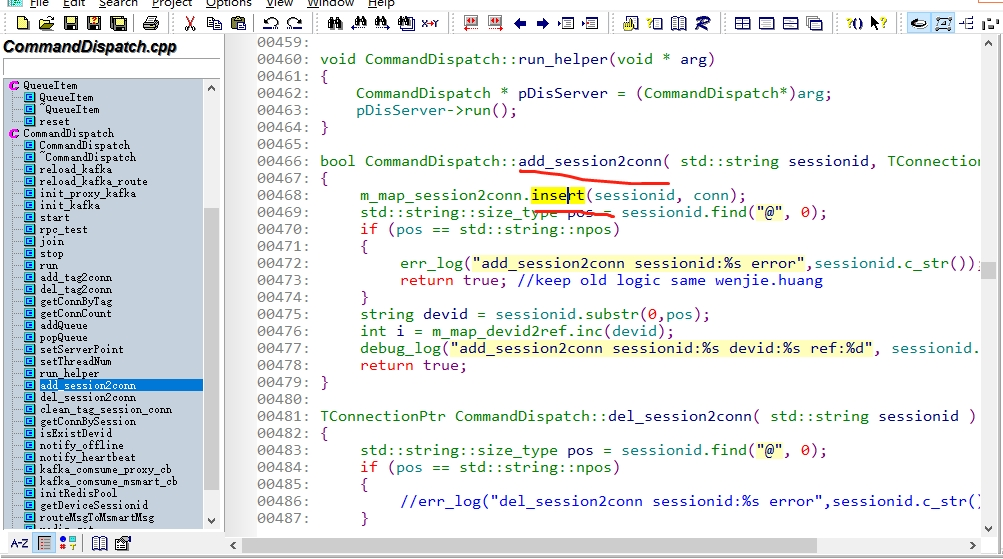
## 学习如何进行线上错误的发生以及错误排查

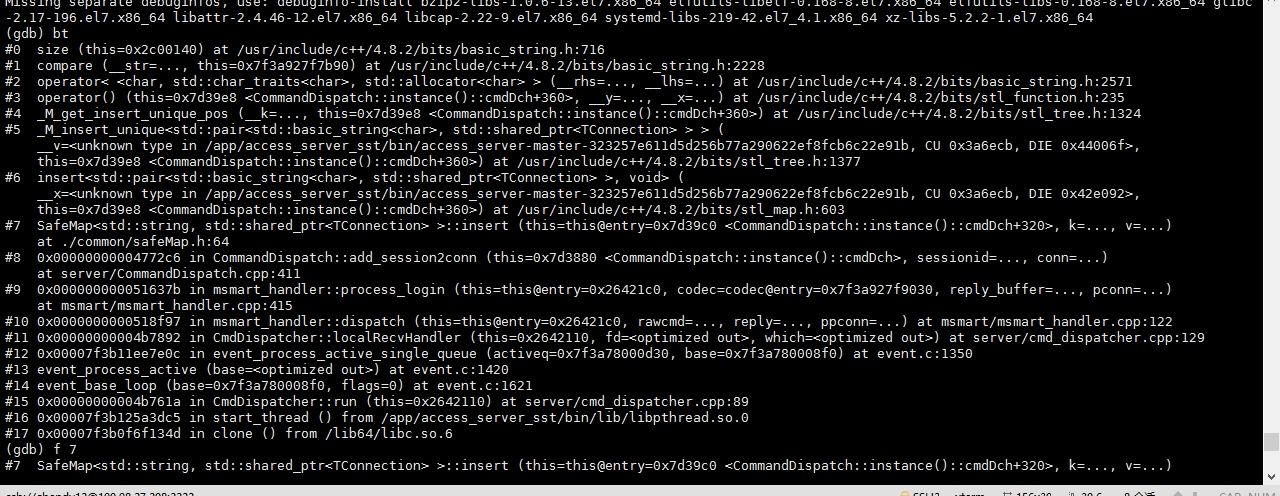
学习了当服务器coredump时如何用gdb结合core文件定位错误，查看错误处相关的堆栈以及值的变化，从而在源代码上作出相应修改，解决错误。

## 协作导师对线上问题的发生进行解决

协助导师解决线上服务器发生的问题，提供自己的思路，让导师进行多种角度的思考，从而更好地解决问题。协助解决的问题如下：

1. 函数运行到aesEncode加密函数或aesDecode解密函数中的AES\_cbc\_encrypt函数时，出现了codedump，
2. 在把sccession id插入map的时候发生了错误，我提出的建议是再认真检查加锁的步骤，保证每一个可能的步骤都要避免竞态而导致的coredump





## 其他产出

a) 为接入层代码的基本框架编写了codeReview 文档，方便自己以后的查看以及后人的理解

b) 为接入层的大部分主要逻辑代码添加了注释，方便自己以后的查看以及后人对代码的理解

c) 编写了go语言的编译环境配置文档，可以方便部门里想要配置go语言环境的同事。

d) 编写了几种类型的coredump分析过程以及改进过程，方便以后的同学学习

# 个人收获和不足

## 个人收获

**技术上：**

* 熟悉了接入层的代码以及其设计框架，学习了libevent、redis、kafka等框架的使用，对多线程并发服务器的理解加深了，为以后的工作做好了铺垫
* 经过自己亲身的压测体验之后，了解了改变各种参数对服务器的影响，自己对服务器的设计理解也加深了
* 学习了一门新的语言——go语言，自己的技术栈加深了
* 学习了如何用gdb结合core文件快速定位到出现coredump的地方，以便更好地解决问题
* 经过协助导师解决线上的coredump问题，自己对线上问题的发生与解决有了初步的理解，大大有助于我以后的工作

**其他：**

* 收获了经验丰富并且耐心的导师指导
* 收获了良好的同事关系
* 收获了美的集团对“美少年”的关怀——美少年运动会、下午茶、团建活动、各种指导课程（如何适应、提升、提高效率等）

## 2.2. 不足

* 熟悉业务、学习需要的技术时还不够快、不够全面，还未能独立承担起接入层的一些工作