


# 服务器百万级并发


---

环境:

image-20240823190928490

**原始并发量:**

比较小

image-20240823192504813

**调整1:**

1. ulimit -a

fd在网络io中就相当于一个文件夹，要把fd扩大

image-20240823192041813

问题：遇到core dump。

遇到段错误，解决方案：

1. 先理清楚代码逻辑，才可能调试
2. 发现可以增加connlist的数组大小；

**调整2:**

image-20240823192257665

问题：并发量增大，仍出现段错误


image-20240823192426193

image-20240823192438831


**调整3:**

直接connlist干到1048576 (doge)

并发量到了六万多，但仍然core dump

image-20240823192831956

原因：

image-20240823193632110

解决方案：

1. 从**sport**入手：使用多个虚拟机，提供多个ip来提供不同的ip: port;
2. 从**dport**入手：使用多个目的端口来接收连接；

#### 调整4:

三台client虚拟机

问题:

有些client 2w多的连接就down了, 有些3w 多

这里说明server已经down了



这里说明不够用了



解决方法:



结果: 确实超过了上面的并发量, 接近20w

#### 调整5:

改变目的端口数



问题:



原因: 开辟数组空间太大, 超过**2G**, 10576是2g, 加上buffer超过了报错

#### 调整6:

改变了BUFFER\_LENGTH = 512

问题:



时间变长了许多;

原因:




客户端不增加了, 停止客户端导致服务器也停止了

#### 调整7:

调整conf的fs-file-max



 image-20240823211416453

问题:

 image-20240823212334109

原因:

操作系统内存管理，当网络io占用超界会有强制CARE的风险。


tcp协议栈的tcp\_mem设置小了

#### 调整8:


conf加入如下配置:

(16g内存的话)

 image-20240823214259027

 image-20240823214342348

bug:

 image-20240823213011266

问题:

并发量还是90w左右上不去

原因: 客户端进程被down了

客户机的内存比较小，这仅剩的一点内存可能不能全部都给client进程使用



 image-20240823214935172

 image-20240823215222552

#### 调整9:

conf的tcp内存 调小


 image-20240823215619689

问题:

还是同样客户端进程被killed，已经触发了tcp内存回收了

#### 调整10:

直接把虚拟机的内存加大一点!

 image-20240823220309653

结果:

96w!!! 客户端killed

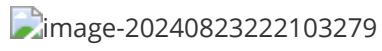


**调整11:**

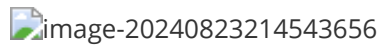
把**tcp\_mem** 默认的值放到最大值 (让tcp使用更大的内存时才回收)

结果:

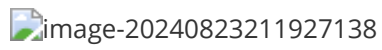
**100W!!!!!!**



测试项目:



思考:



1.超大数组优化: 可以用**动态数组**

