线上问题分析: The target server failed to respond (目标服务器返回失败)

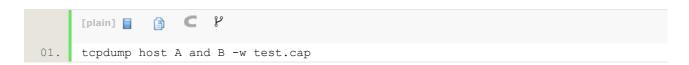
背景

后台服务<u>通过REST接口调用第三方服务</u>,<mark>调用偶尔会发生失败</mark>。查看后台日志发现,抛出The target server failed to respond。

该问题只是在特定操作下才会出现,暂时没有任何规律可言。

问题分析

- 1、<u>到第三方服务器</u>上查看后台日志,发现在相同时间点没有日志输出。查看tomcat的locahost_access访<mark>问日志</mark>,也没有发现任何日志信息。初步怀疑<mark>请求没有达到第三方服务器</mark>。
- 2、怀疑是gateway的问题,但是调用服务是通过ip直连的方式,没有通过gateway中转。排除此种可能。
- 3、<u>为了验证http请求是否发出,以及是否到达了第三方服务器,使用tcpdump</u>工具,对网络包进行了监 控。只监控这两台服务器,脚本为:



注意,此处生成的是cap文件,方便在windows上用wireshark工具进行分析。

用wireshark分析的结果如下:



从分析结果可以看出,<u>已经与第三方服务器建立了连接,但是</u>第三方服务器没有返回HTTP应答,只有一个 TCP响应报文。由此可以排除步骤一的疑点。

但是<u>按顺序分析这几个报文,一开始就建立一个HTTP连接,不太符合TCP连接的规范。理论上,应该是走</u>TCP的三次握手。有一种可能是,<u>使用了连接池</u>的机制,重用了TCP的连接。但是<u>为什么第三方服务器会发送一个重置的TCP报文</u>呢?

4、在我们的服务器上查看连接情况,使用netstat -nltpa,发现有个CLOSE_WAIT状态的连接:

咱们的<u>HTTP协议使用的是1.1版本</u>,因此<u>会默认为Keep-Alive,在<mark>请求头</mark>中,<mark>Connetion</mark>确实设置为了 <u>Keep-Alive</u>。<u>排除客户端主动关闭</u>的可能。</u>

第1页 共2页 2017/5/12 下午7:39

∃ Hypertext Transfer Protocol
⊕ GET /pricecenter/v1/price/mcondit

Host: 10.77.145.107:10072\r\n Connection: Keep-Alive\r\n

客户端出现CLOSE_WAIT状态的TCP连接

查看<u>第三方服务器的<mark>tomcat配置</mark>,发现<mark>超时时间是20s</mark>。因为在请求处理完后的20s,<mark>服务器端</mark>就会主动关 闭TCP连接。</u>

有了这个线索,我们就掌握了<mark>重现的规律</mark>:<u>发送一个请求后20s,待客户端连接状态变为CLOSE_WAIT状</u>态,再发送一次请求就出现错误了。

结论

客户端肯定会出现CLOSE_WAIT状态的TCP连接

- 1、客户端使用了HttpClient的连接池机制,因此TCP连接会被重复使用,并且由于默认使用的是HTTP 1.1 协议,Keep-Alive会被置上,因此不会主动关闭连接。但是由于第三方服务器配置了20s的超时时间,TCP连接会被关闭,但是此时客户端的TCP连接会被回收到池子里,不会理睬服务器发送过来的关闭请求,因此客户端的连接状态会变成CLOSE_WAIT,而服务器经过一段时间等待后会关闭自身的TCP连接。如果客户端下一次请求正好使用了这个TCP连接,就会出现服务器返回空应答的情况,从而抛出该问题。
- 2、HTTP协议是基于TCP连接的,如果是<u>重用已有的TCP连接,则TCP三次握手不会发生。抛出该问题,</u>是HTTPClient在解析HTTP头时,发现没有数据。根本原因是因为没有返回HTTP数据包,而是返回了TCP数据包。 问题根源:

总结

- 1、可以使用HttpClient重试机制,调用第一次失败时,第二次可能就正常了。HTTP 1.1默认使用了<mark>持久连接机制,像这种情况不可避免。如果访问频繁</mark>的话,该问题不会发生,而且性能会比较高。
- 2、开启一个线程,<u>定期关闭<mark>无效的TCP连接</u>。</u></mark>

HttpClientConnectionManager.closeExpiredConnections()、

HttpClientConnectionManager.closeIdleConnections()。这个方法没试过。(此方法为<u>官方文档的推荐方</u>法)

- 3、在请求头里设置"Connection":"close"。<mark>弊端</mark>是,没有用上连接池的性能。
- 4、设置HttpClient的<mark>连接重用策略</mark>。目前的重用策略,对于HTTP 1.1,都是默认为可以重用。

服务端

如果服务器使用长连接配置,那不做调整应该也不会碰到该问题。

但是服务器如果设置了keep-alive时间,客户端如果也同样设置,感觉不太灵活。

有没有更好的方法,还需要去研究下HttpClient的实现。

第2页 共2页 2017/5/12 下午7:39