# **motan-2：motan的简约限流/熔断方式**

****半原创/参与解决，根据同事文档编排；微信公众号：千里行走；****

****受限图片大小限制，有些图片不是很清晰，可以到微信公众号查看；****

**目录**

**(1).背景**

**(2).原因**

**(3).模拟重现**

**(4).源码解析**

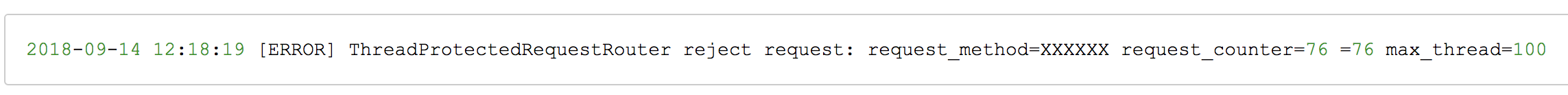
**(5).解决方案**

**(6).总结**

**正文**

**(1).背景**

**线上服务由于调用量及qps都较高，在上线期间，motan日志打出如下错误：**

****

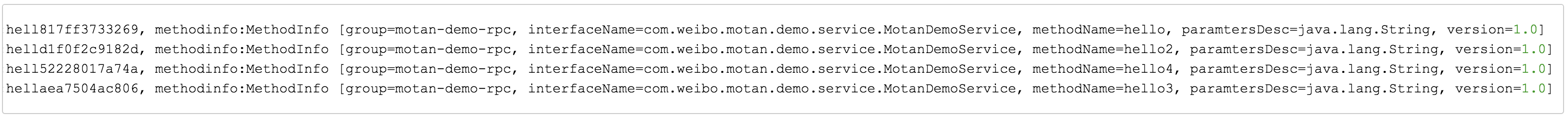
1. **.原因**

**github上作者的回复：**

**<https://github.com/weibocom/motan/issues/551>  
**

# **(3).模拟重现**

**配置motan服务MotanDemoService，提供4个方法，其中hello1的处理逻辑为sleep 1s**

****

**服务配置：**

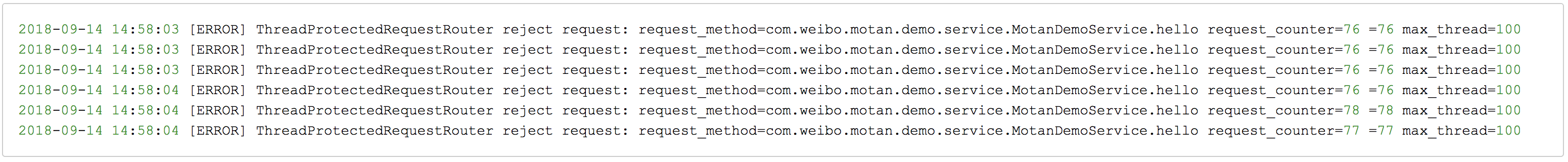
****

**motan服务暴露在8002端口，启动一个客户端，请求服务器1000次**

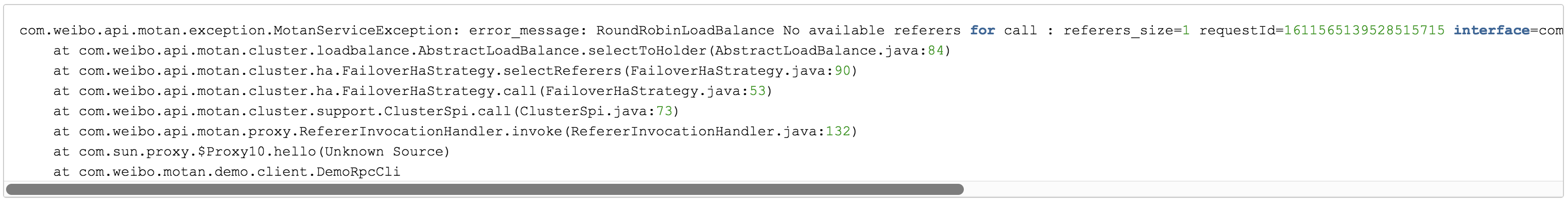
**汇总结果如下：**

****测试场景1**  
 75并发访问服务端，请求1000次，结果如下：  
 motan demo is finish. success: 1000 error: 0**

****测试场景2**  
 80并发访问服务端，请求1000次，结果如下：  
 motan demo is finish. success: 150 **error: 850**  
 **服务端错误信息：****

****

**综述，跟作者描述一致，当并发度达到 方法数的 3/4 \* 100（线程数）= 75时，触发motan的熔断机制，产生大量拒绝请求，客户端报错如下：**

****

# **(4).源码解析**

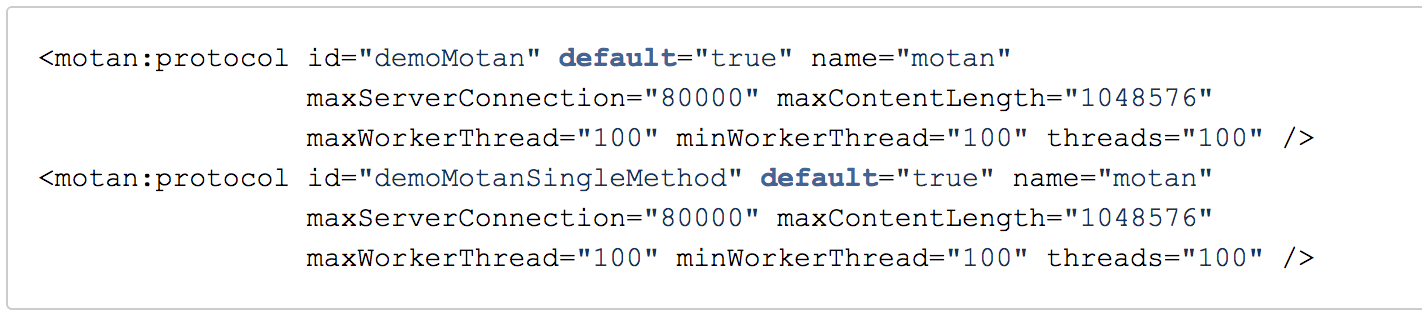
**类ProviderProtectedMessageRouter**

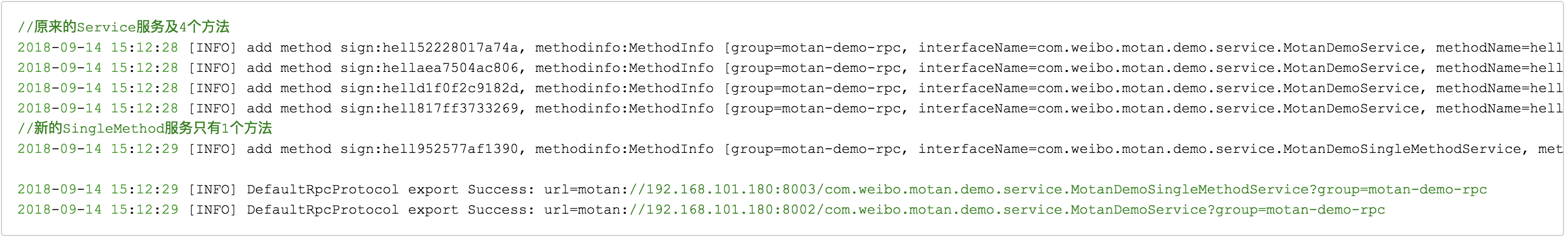
****

# **.解决方案**

**1.自定义ProviderMessageRouter**

**2.通过配置解决  
根据motan的线程池分配为每个protocol配置下一个独立线程池，基于这样的思路，将MotanDemoService服务中访问量最大的接口hello独立暴露一个motan服务， 单独注册一个motan protocol为demoMotanSingleMethod与原来的demoMotan并存，配置信息如下：**

**  
 改造后的motan服务如下：**

****

**客户端将请求指向MotanDemoSingleMethodService的hello方法；**

****期望接口能够承受100并发**，测试结果如下：**

**motan demo is finish. **success: 1000** error: 0**

****修改并发数为150，期望服务器触发熔断，拒绝部分请求**，测试结果如下：**

**motan demo is finish. **success: 1000** error: 0**

****实际与预期不符，推测motan在这种情况下，可能有服务端排队机制，或者客户端阻塞等待请求了**，为了避免在服务端处理能力达到极限时hang死客户端，客户端应该设置合理的超时退出时间，另外也需考虑熔断机制避免引发雪崩。**

# **(6).总结**

**motan默认的保护机制确实带来了一定的困扰，但是motan的这种机制也默认为大家的服务 进行了一种保护，在某个接口并发压力大的情况下依然可以预留25%的线程处理其余的接口 方法，所以大家在设置motan的线程数时要额外注意，是否接口中各个方法的访问压力严重 不均衡，如果发生了上述的错误，则说明这个接口需要单独抽离出来了**

**深入研究各种极端条件下，motan的排队，拒绝行为，找到设置合理motan配置的最佳实践**