**服务器环境信息(先启动Zookeeper)：**

1：Gateway项目，NIOmaxThreads=500,port=9003,启动参数:-server -Xms128M -Xmx256M  
2：Order项目(两台机器各一个应用)，NIOmaxThreads=500,port=9002,启动参数:-server -Xms128M -Xmx256M  
3: Product项目，NIOmaxThreads=500,port=9001,启动参数:-server -Xms128M -Xmx256M  
4: Hystrix DashBoard查看服务

5: 压力测试机器服务一台

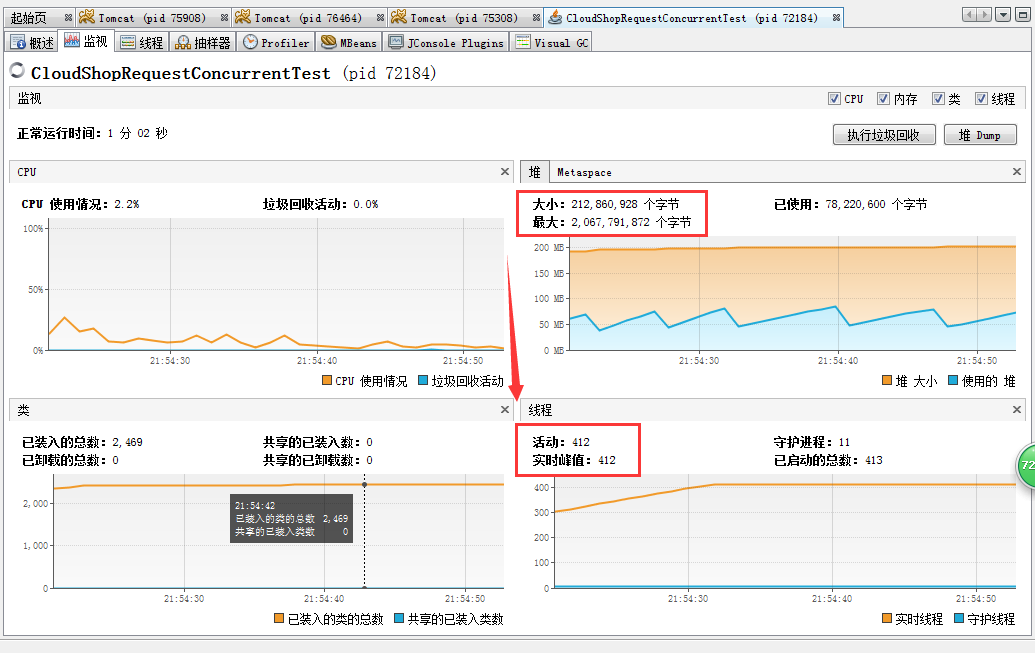
6: 压力测试主代码目录：CloudShopRequestConcurrentTest.java当中

**场景一：400线程并发循环请求30次----RestTemplate直接访问方式**

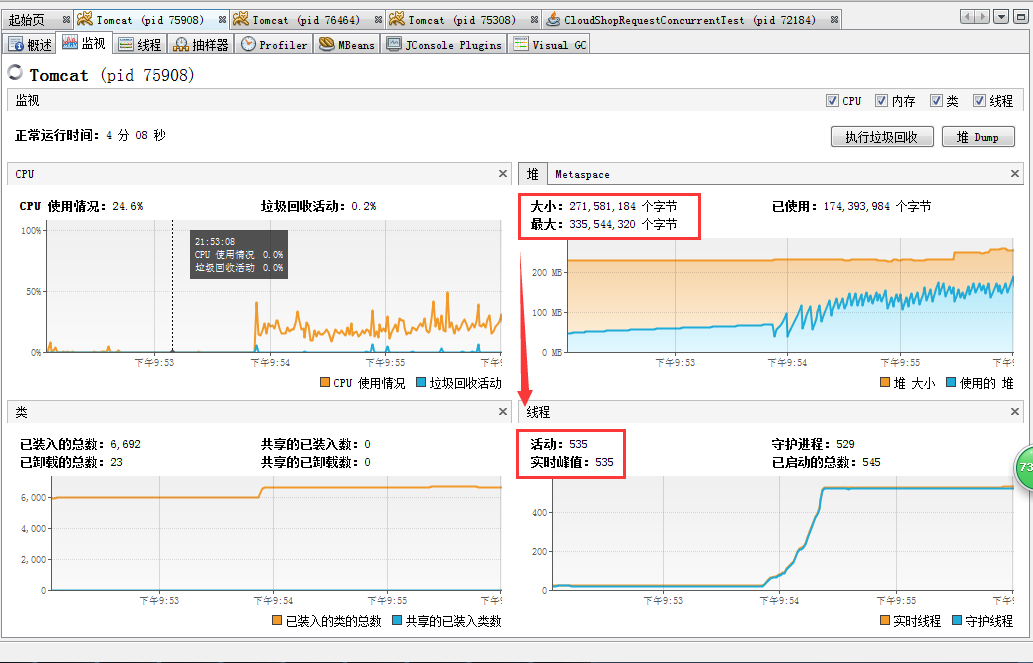
压力测试方法描述

用代码启动400个线程，每个线程对Gateway项目做并发请求，两种请求：1）RestTemplate直接访问服务方式；2）Hystrix安全请求方式

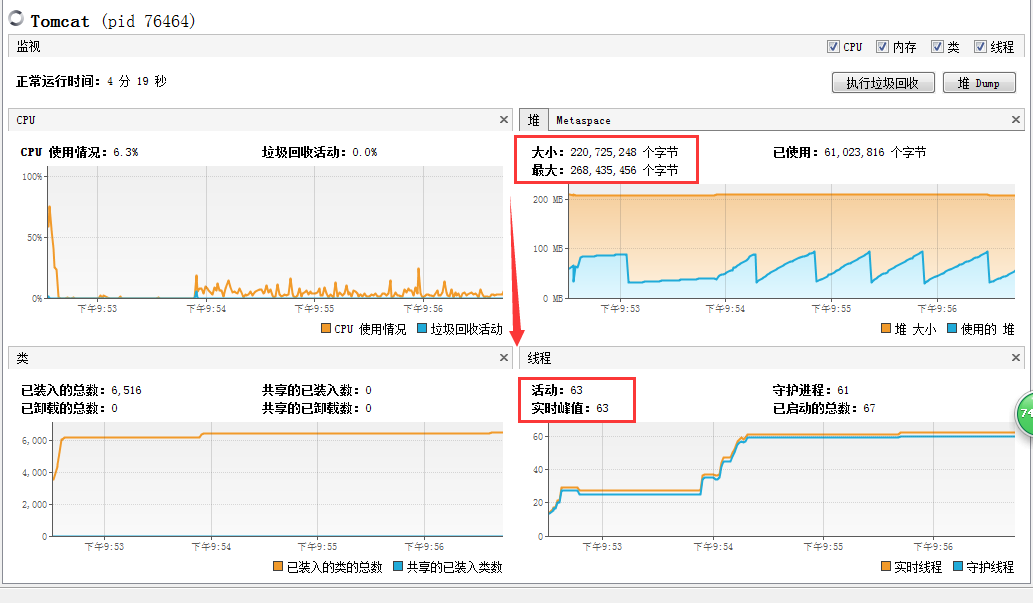
压力机(理论接口请求48000次)



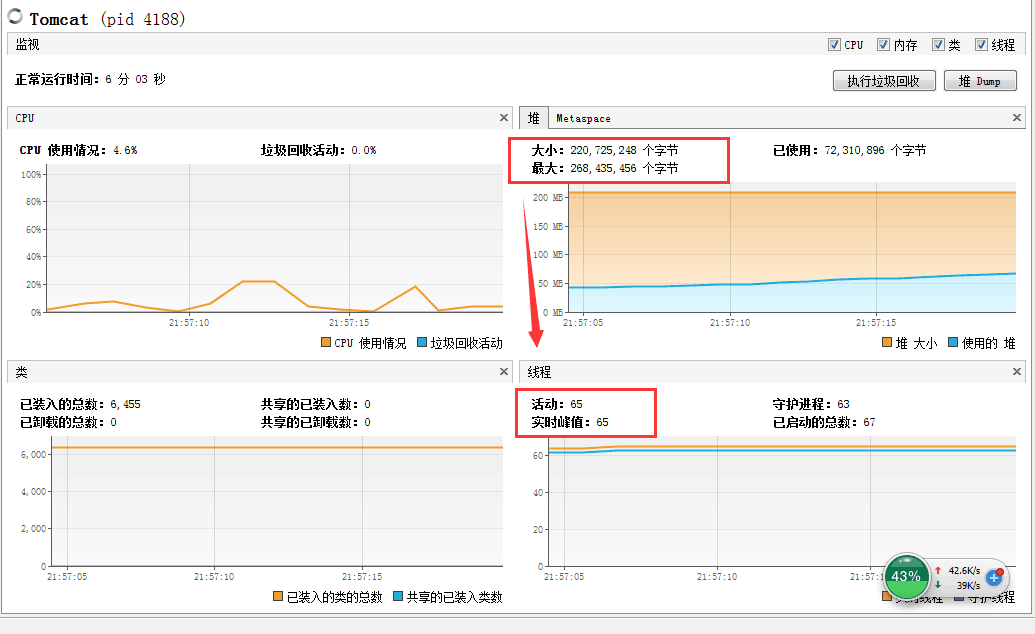
Gateway项目 (共处理接口17881)



Order服务A(本地) (共处理接口8940)



Order服务B(远程) (共处理接口8940)



压力测试结果:

1：压力结束之后20分钟Gateway项目还是在忙，线程下不来，估计是在TomcatNIO模式的等待处理队列工作了

2：压力测试机理论接口发出48000次，而实际GateWay项目请求了17881次，两个Order项目请求集合值刚好等于此。那差异的接口次数去哪里了？？

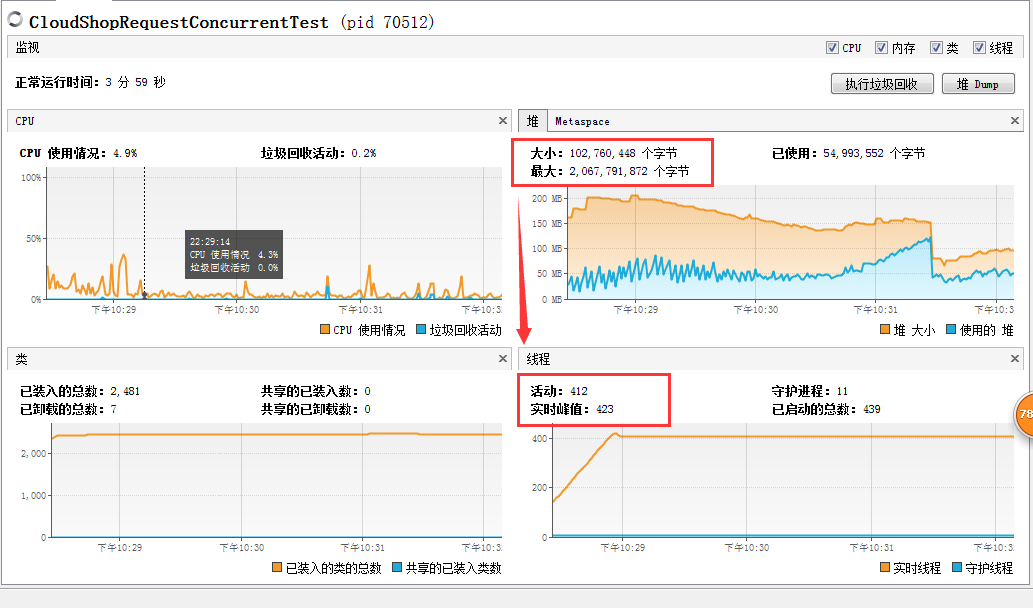
3：在压测过程中，通过其他渠道请求实际Gateway项目接口，需要等待很长很长很长时间(70S+)才有响应。

**场景一：400线程并发循环请求30次----Hystrix安全方式方式**

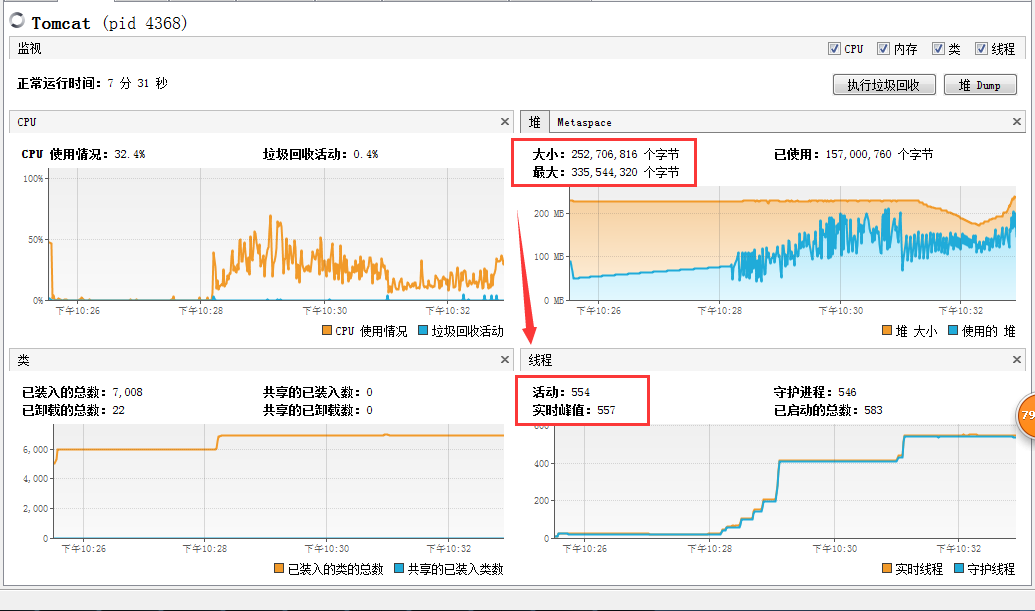
压力测试方法描述

用代码启动400个线程，每个线程对Gateway项目做并发请求，两种请求：1）RestTemplate直接访问服务方式；2）Hystrix安全请求方式

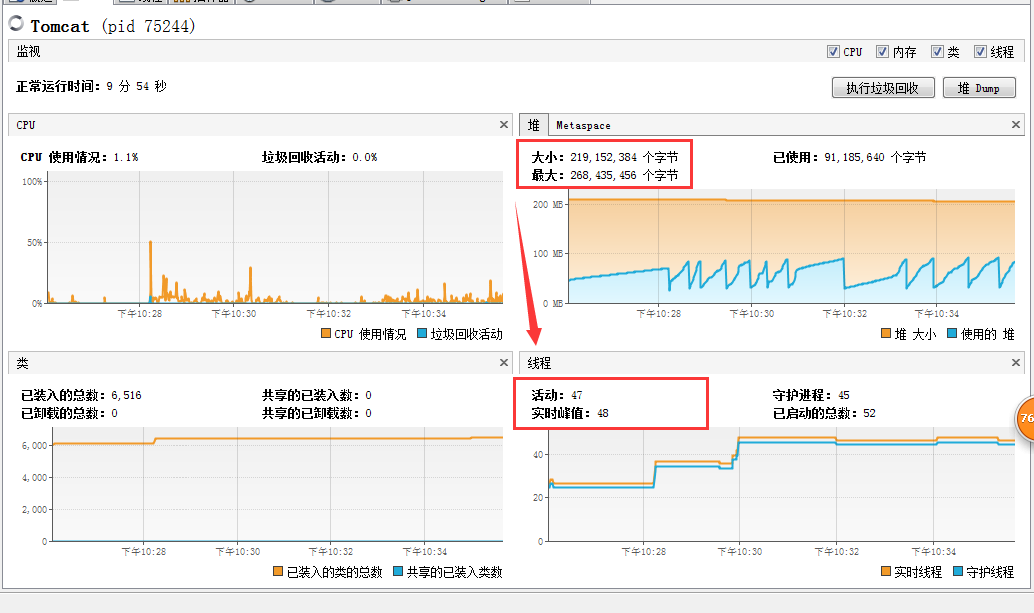
压力机(理论接口请求60200次)



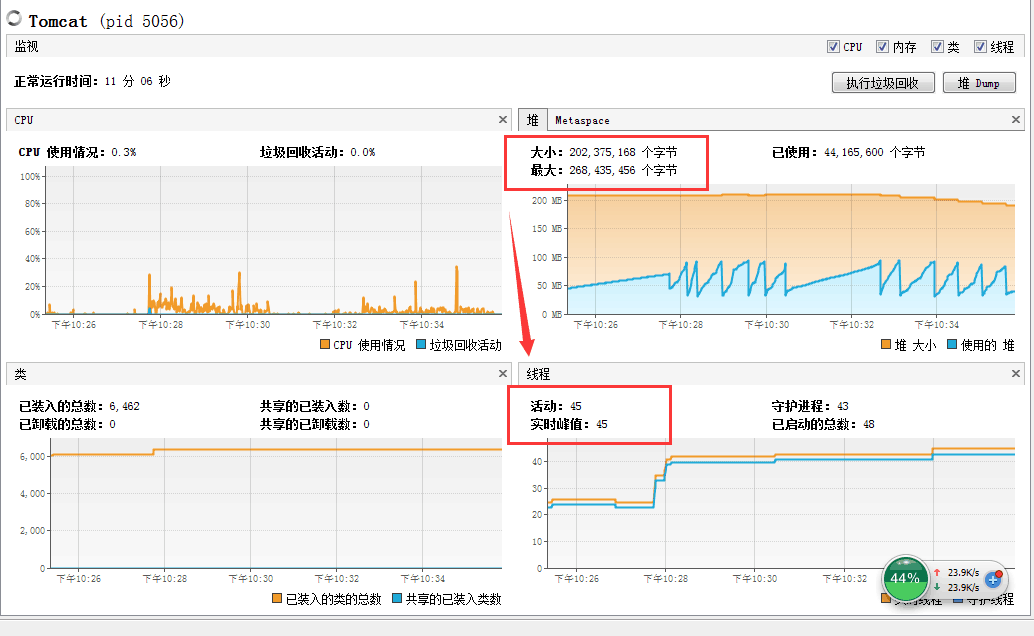
Gateway项目 (共处理接口60203)



Order服务A(本地) (共处理接口4612)



Order服务B(远程) (共处理接口4703)



Product服务A(本地) (共处理接口9712)

压力测试结果:

1：压力结束后，Gateway项目立即空闲下来。

2：压力测试机理论接口发出60200次，而实际GateWay项目请求了60203次，两个Order项目+Product项目共调用19027次，差异的41196是做了熔断处理的结果，没有差异。但是这么多熔断接口怎么处理(由于是模拟了异常失败的接口，导致熔断比较容易打开)

3：在压测过程中，通过其他渠道请求实际Gateway项目接口，很快就有响应

**场景一：500线程并发循环请求50次----总结说明**

这种压力测试方式欠妥，表现上看是RestTemplate直接访问服务效果比Hystrix安全访问服务方式效果更好，其实这里面有误区，因为RestTemplate直接访问的时候，很多线程是出于等于的状态，而Hystrix会有超时中断，所以在单位时间内，实际上Hystrix方式的请求吞吐量要比RestTemplate高很多。

**场景二：线程池设置多少为合适。**

经过测试，发现线程池大小设置为10，250个并发的时候，已经出现线程池满的情况。

在实际项目中，线程池是通过一个组公用的，所以可以根据实际情况设置不同，如少用工程可以设置少一些，用的多的工程可以设置大一些。经过调整线程池设置为20的时候，250个并发没有线程池满的情况。

**断路器开启或者关闭的条件(官方说法)**

*1、 当满足一定的阀值的时候（默认10秒内超过20个请求次数）  
2、 当失败率达到一定的时候（默认10秒内超过50%的请求失败）  
3、 到达以上阀值，断路器将会开启  
4、 当开启的时候，所有请求都不会进行转发  
5、 一段时间之后（默认是5秒），这个时候断路器是半开状态，会让其中一个请求进行转发。如果成功，断路器会关闭，若失败，继续开启。重复4和5。*