

1.测试目的

让开发者对hero_mall项目的性能、负载和容量有个准确的认知。预先发现并解决性能问题，能够更好把控业务系统质量。

2.测试功能

- *压测工具Jmeter
- *可视化监控平台Grafana

3.系统服务环境

- 业务服务
 - 服务器系统：centos7系统
 - 内存：4G8G
 - 部署方式：单机部署
- 数据库
 - 服务器系统：centos7系统
 - 内存：4G8G
 - 数据库版本：5.7
 - 数据库名称：hero_mall_one

4.业务服务启动参数

```
export JAVA_HOME=/usr/local/hero/jdk1.8.0_261
export JRE_HOME= JAVA_HOME/jreexportCLASSPATH = . : {JAVA_HOME}/lib: JRE_HOME/libexportPATH =
{JAVA_HOME}/bin:$PATH

#=====

# init

#=====

export SERVER="hero_web"
export JAVA_HOME
export JAVA="$JAVA_HOME/bin/java"

# 获取当前目录
```

```
export BASE_DIR= cd $(dirname $0)/.; pwd
```

```
# 默认加载路径
```

```
#export DEFAULT_SEARCH_LOCATIONS="classpath:/,classpath:/config/,file:/,file:/config/"
export DEFAULT_SEARCH_LOCATIONS="file:/,file:/config/"
```

```
# 自定义默认加载配置文件路径
```

```
export CUSTOM_SEARCH_LOCATIONS= DEFAULT_SEARCH_LOCATIONS, file : {BASE_DIR}/conf/
```

```
#=====
```

```
# JVM Configuration
```

```
#=====
```

```
JAVA_OPT=" JAVA_OPT - server - Xms512m - Xmx512m - Xmn256 - XX : MetaspaceSize = 128m - XX : MaxMetaspaceSize = 320m" JAVA_OPT = " {JAVA_OPT} -XX:-OmitStackTraceInFastThrow -
XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath= BASE_DIR/logs/java_heapdump.hprof" JAVA_OPT = "
{JAVA_OPT} -XX:-UseLargePages"
```

```
JAVA_OPT=" JAVA_OPT - jar {BASE_DIR}/ SERVER - *.jar" JAVA_OPT = " {JAVA_OPT}
JAVA_OPT_EXT" JAVA_OPT = " {JAVA_OPT} --spring.config.location=${CUSTOM_SEARCH_LOCATIONS}"
```

```
# 创建日志文件目录
```

```
if [ ! -d " BASE_DIR/logs" ]; then mkdir {BASE_DIR}/logs
fi
```

```
# 输出变量
```

```
echo " JAVA {JAVA_OPT}"
```

```
# 检查start.out日志输出文件
```

```
if [ ! -f " BASE_DIR/logs/ {SERVER}.out" ]; then
touch " BASE_DIR/logs/ {SERVER}.out"
fi
```

```
#=====
```

```
# 启动服务
```

```
#=====
```

```
# 启动服务
```

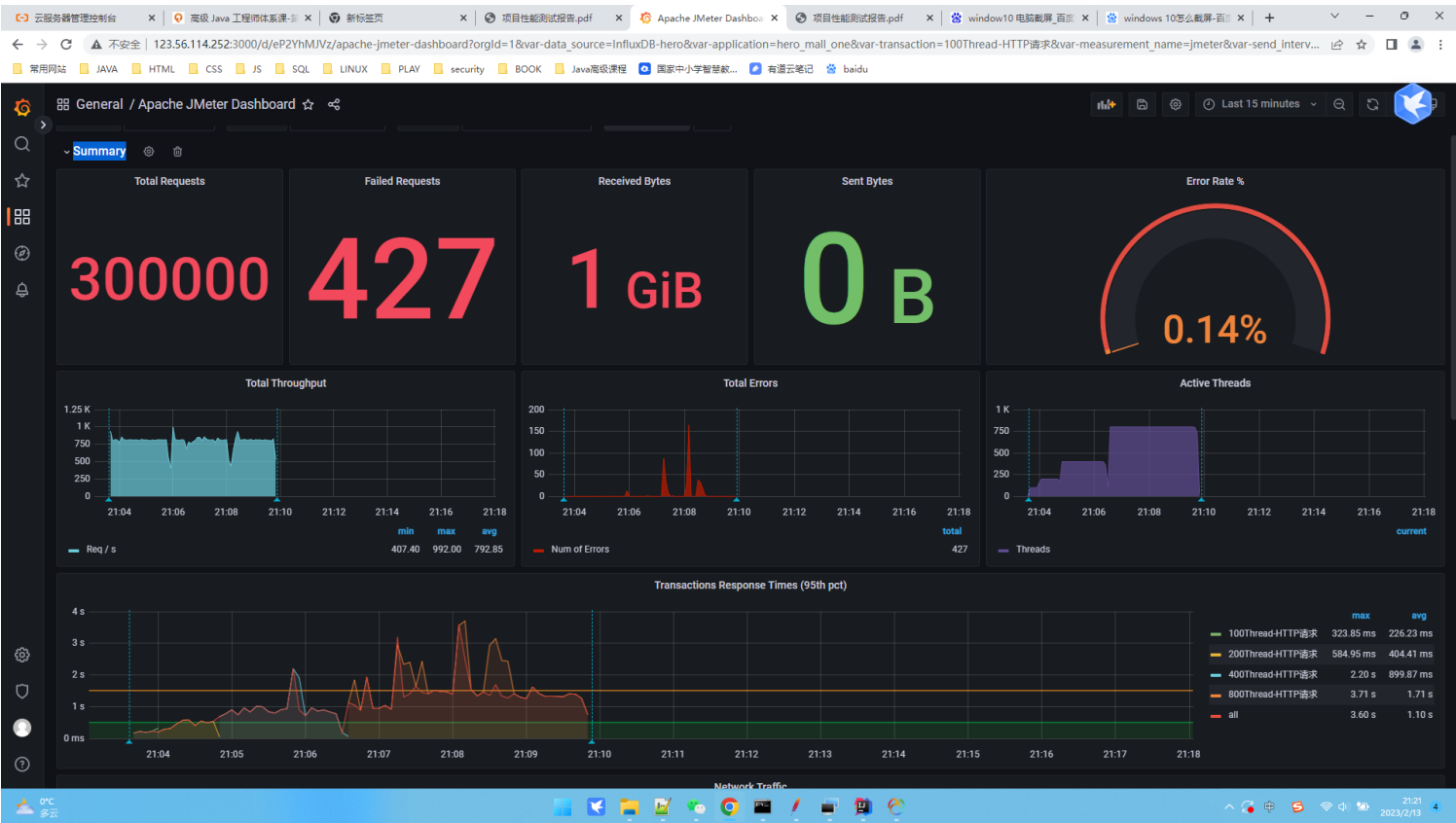
```
echo " JAVA {JAVA_OPT}" > BASE_DIR/logs/ {SERVER}.out 2>&1 &
nohup JAVA {JAVA_OPT} hero_web.hero_web >> BASE_DIR/logs/ {SERVER}.out 2>&1 &
echo "server is starting, you can check the BASE_DIR/logs/ {SERVER}.out"
```

5.测试场景

验证hero_mall服务获取商品信息接口在不同并发规模的表现

用户访问接口并发逐渐增加的过程。线程梯度：100、200、400、800个线程，每个梯度执行200次。

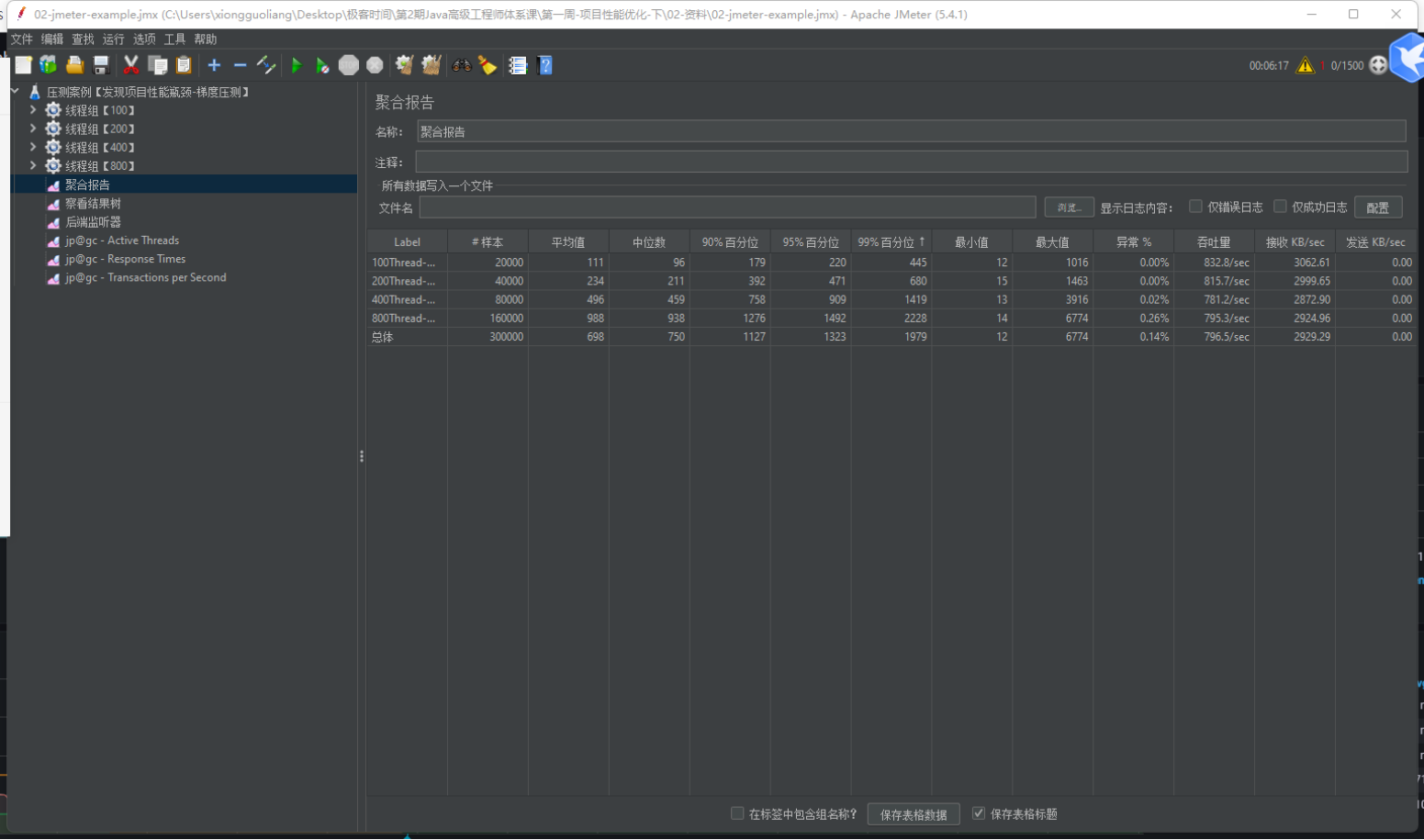
*grafana综合展示图.png



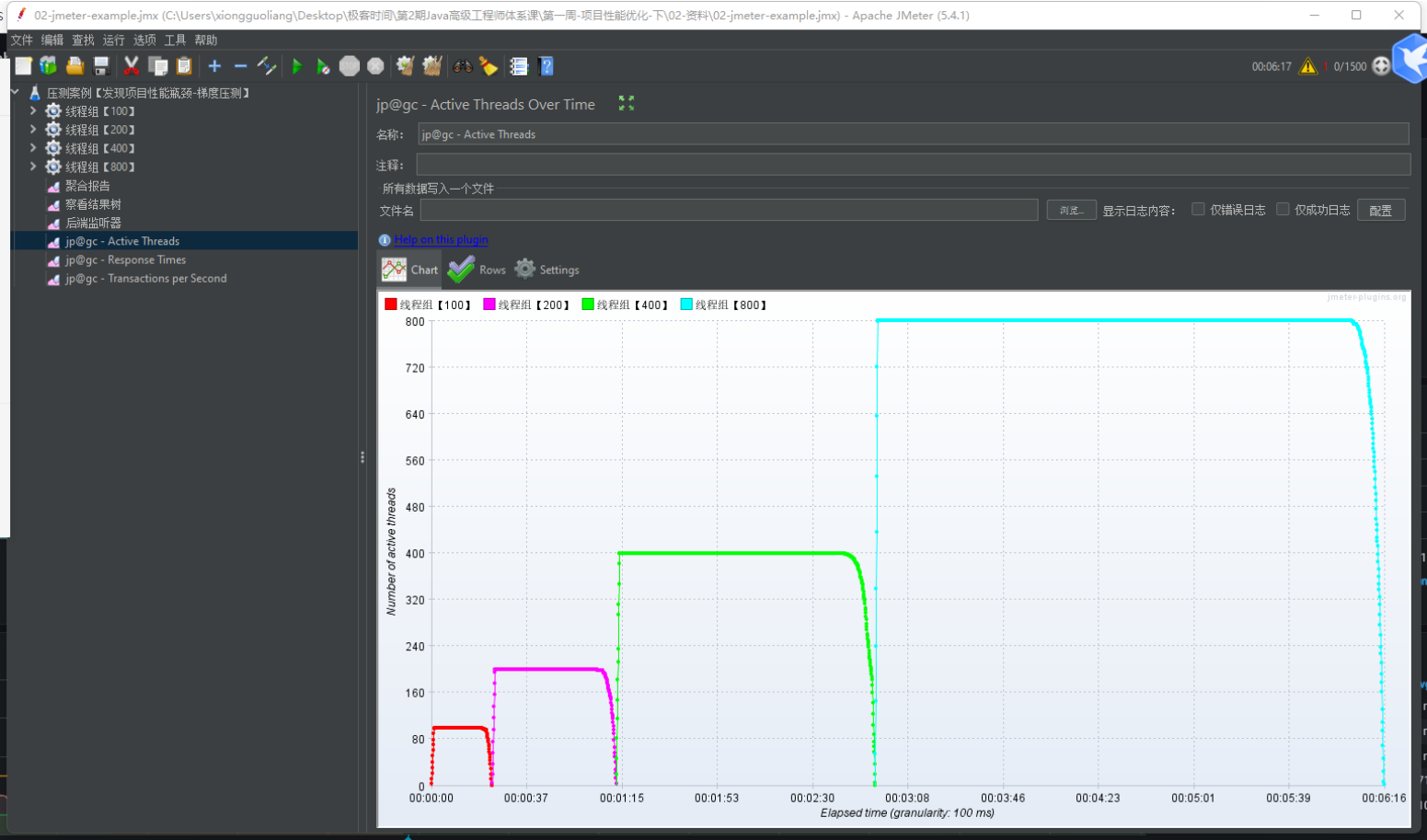
*grafana服务器性能指标.png



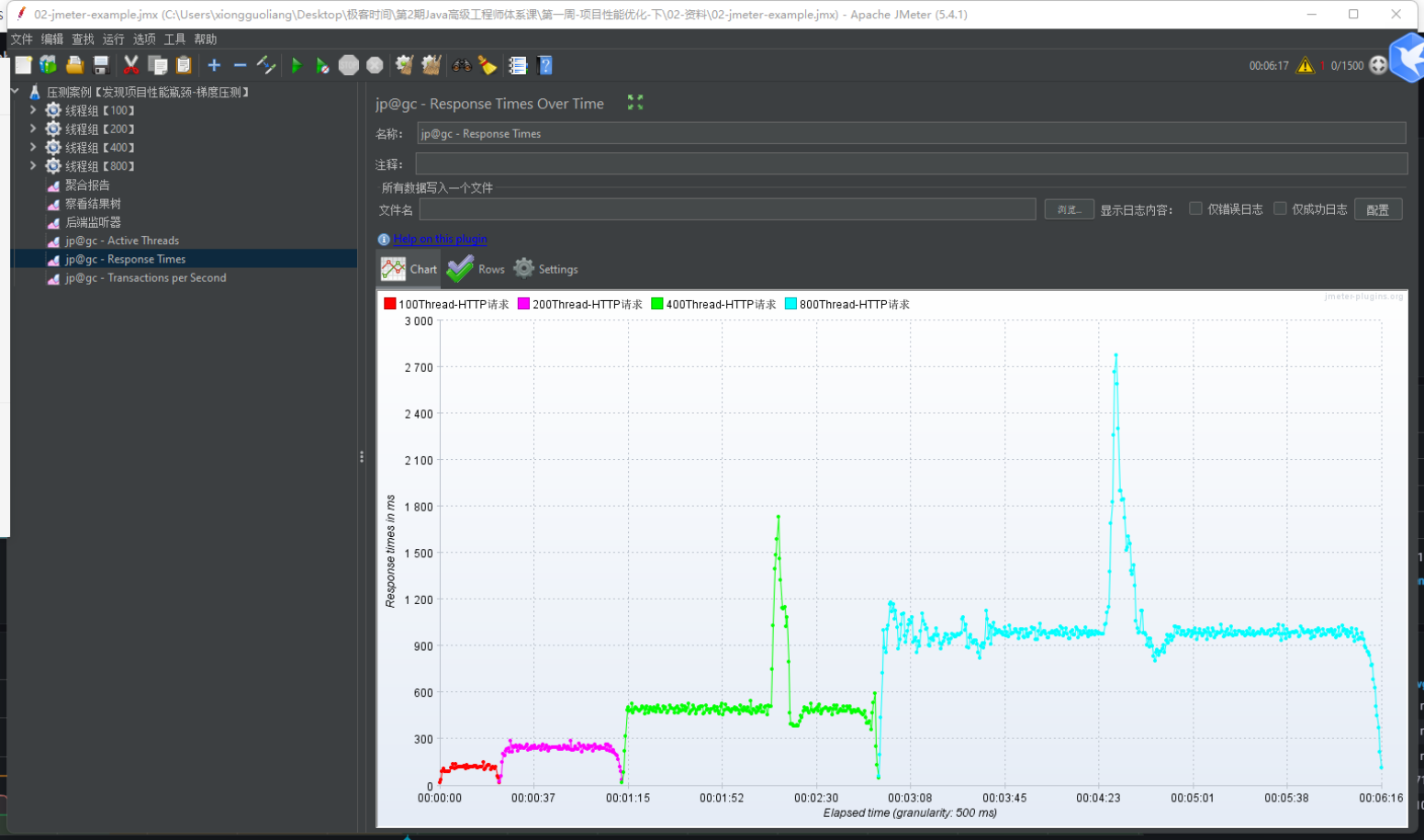
*jmeter聚合报告图.png



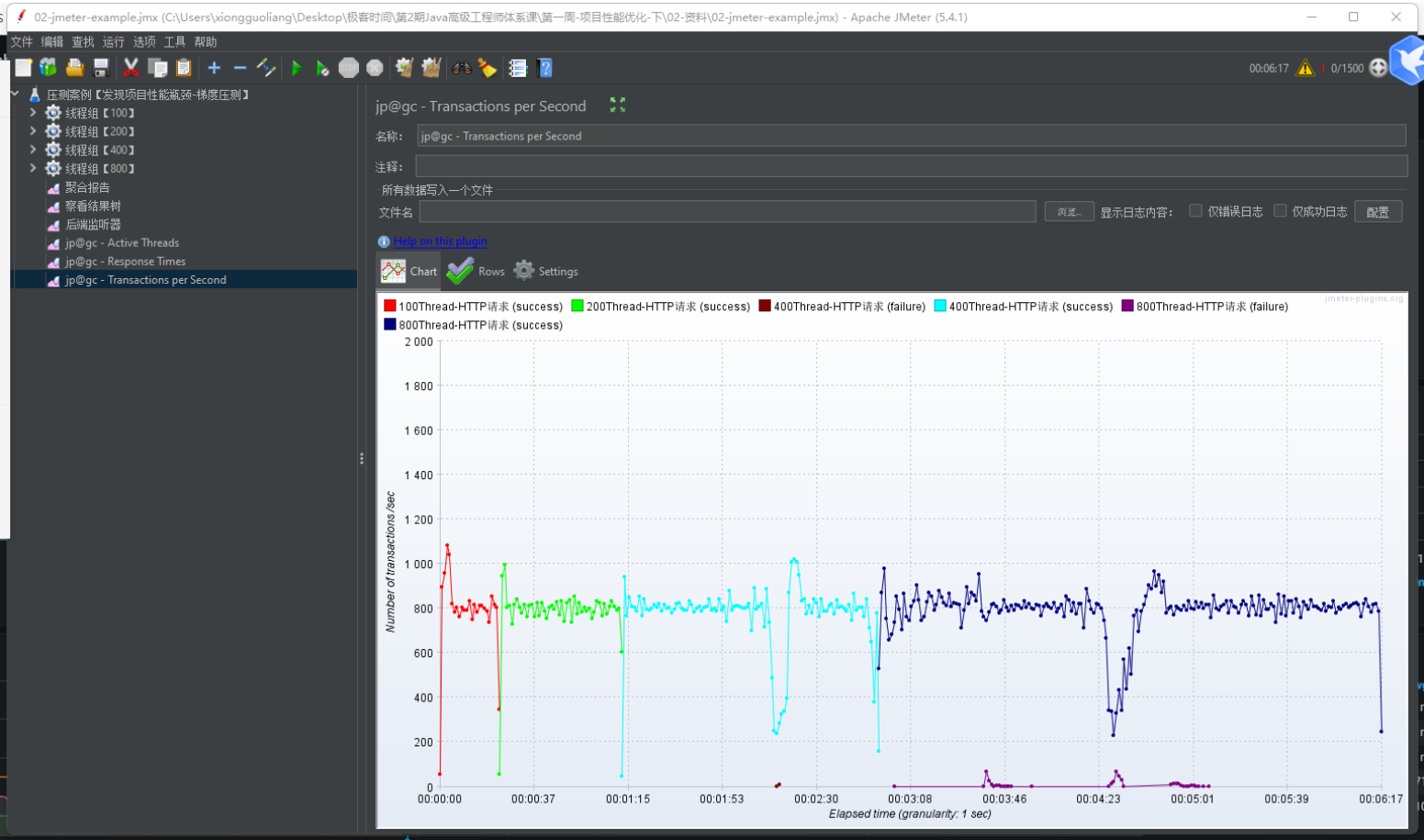
*jmeter存活线程图.png



*jmeter平均响应时间图.png



*jmeter吞吐量图.png



6.测试报告分析

随着线程梯度不断增加有如下情况。当线程成倍增长的时候，响应时长也随着成倍增长，超过三秒的接口变多，失败率较之前有明显的增加。

- 1.吞吐量基本维持在800/sec，较稳定；
- 2.网络带宽基本保持在8Mb/s，较稳定，说明带宽不是影响因素。
- 3.cpu使用率不高且较为稳定，说明服务器性能不是影响因素。

7.优化建议

tomcat自身所能支持的最大并发数默认为200，可适当的调整tomcat的最大线程并发数。或者使用更优的网络模型NIO或者AIO。

以上仅供参考，如有不足或者偏差，请指正。