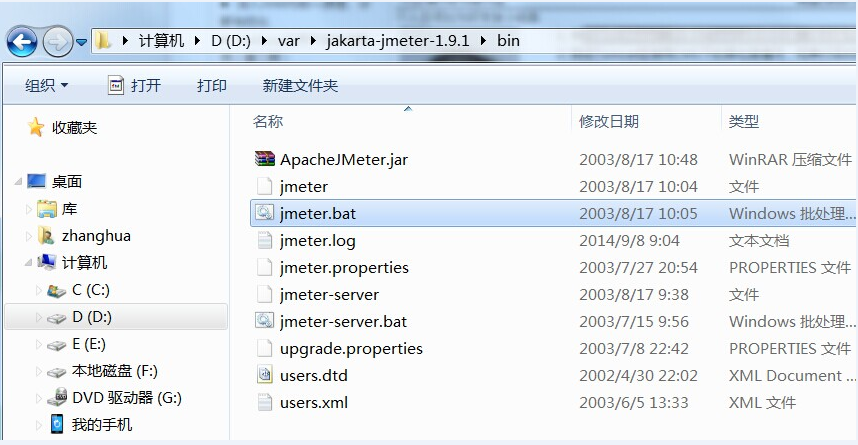
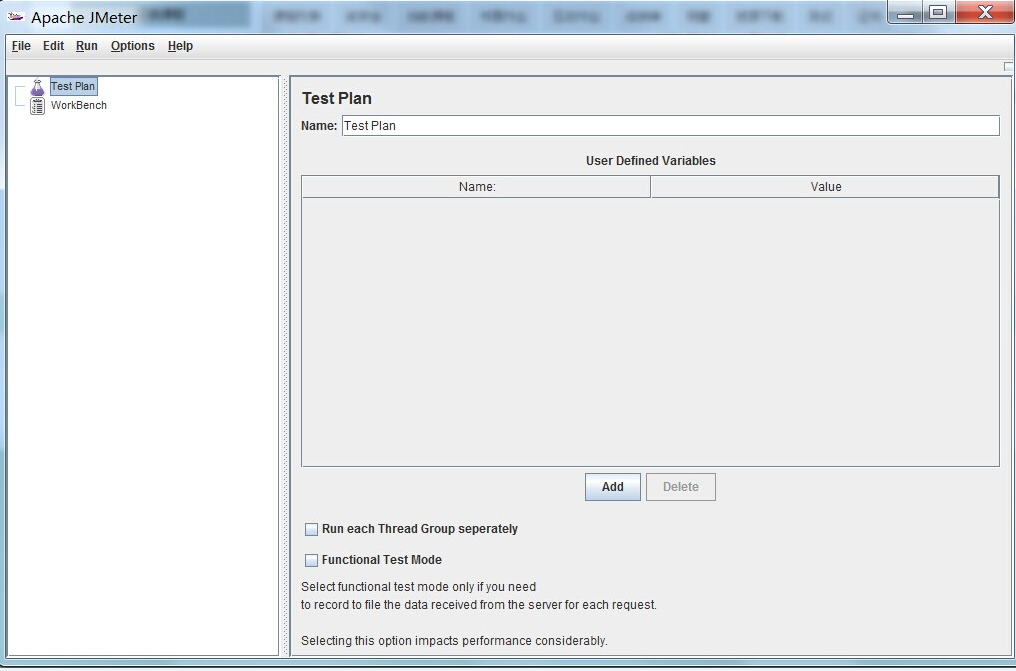
1. 学习使用JMeter，并测试Tomcat的吞吐率，给出有关配置和测试结果的至少5个截图

答：我的jmeter和tomcat都在同一台机器

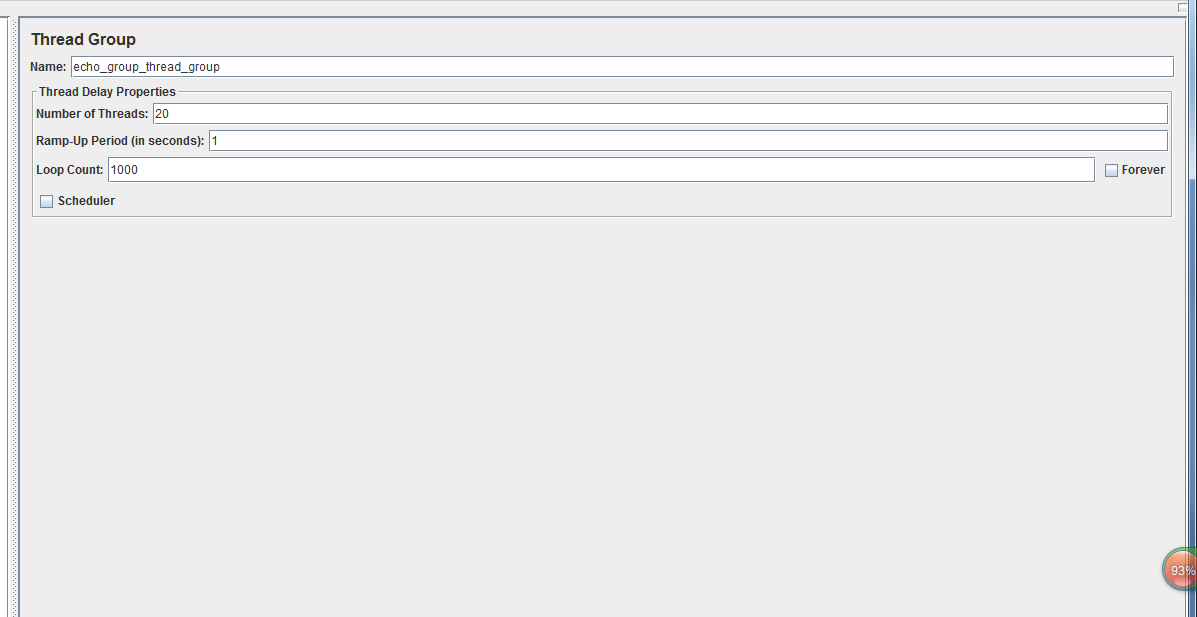
1、首先要安装jmeter,安装过程非常简单，将下载好的zip包解压，并配置环境变量就可以了，此时点击是jmeter就可以运行

****

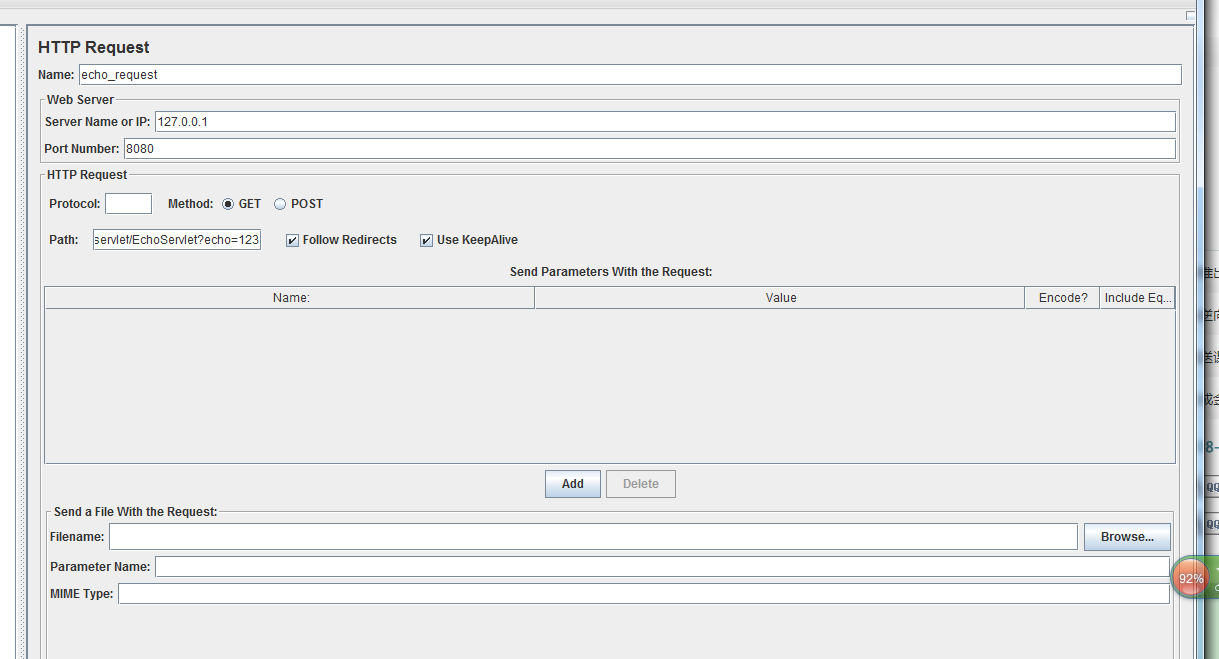
**2、运行Jmeter.bat：**

****

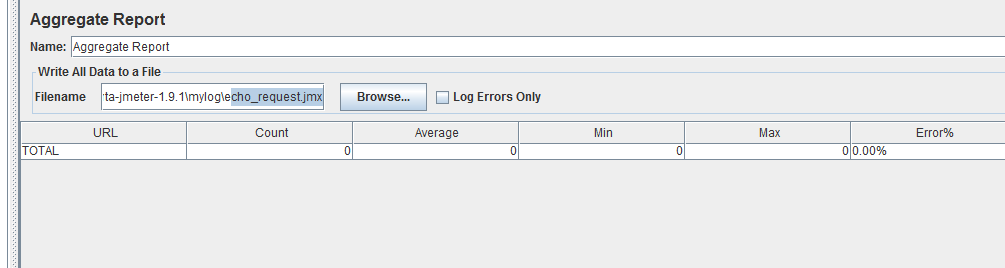
**3、增加Jmeter线程组**

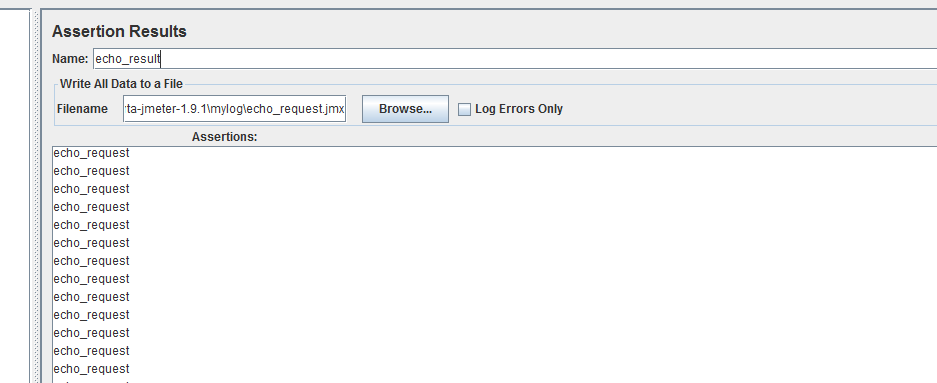
****

**4、设置http request**

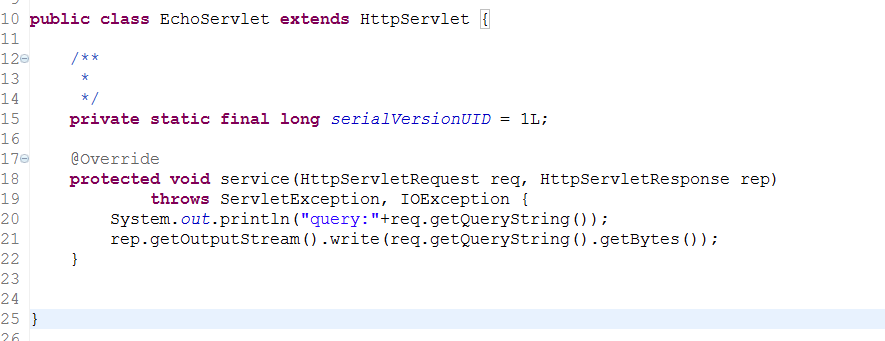
****

**5、增加聚合报表和聚合的result：**

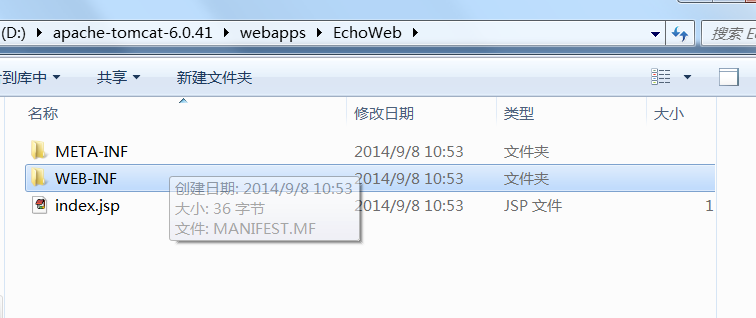
****

****

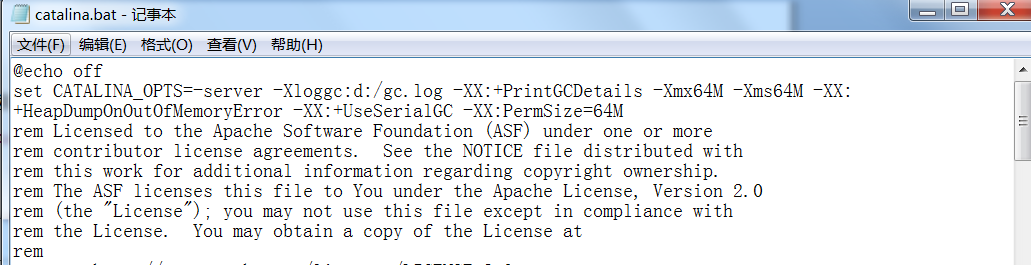
**6、开发echo servlet，代码如下：**

****

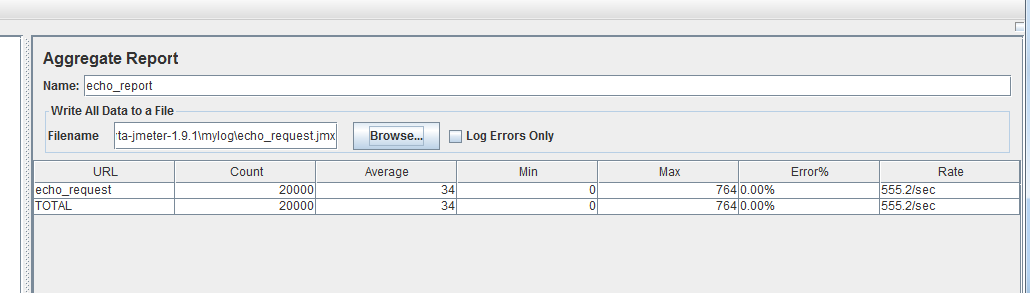
**7、配置web.xm并发包到tomcat下面：**

****

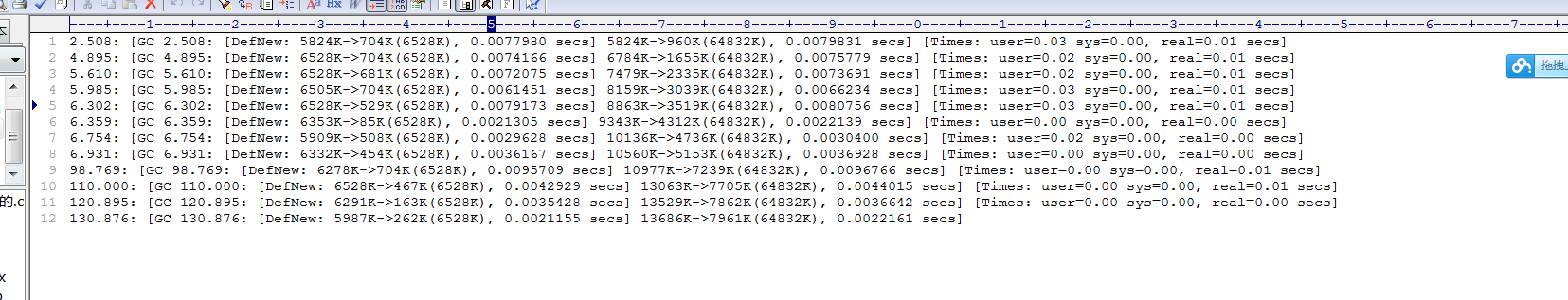
**8、在catalina里面设置jvm启动参数：我初始设置比老师大，gcLog存放d盘**

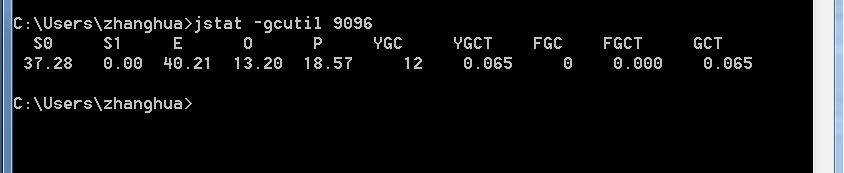
****

**9、启动tomcat,并运行测试用例，查看聚合结果：可以看到Min最在0，最大在764ms,我想因为是本机所以min可以达到0ms,大概每秒处理550，没有发生错误**

****

**10、看gc.log日志：因为我这个是servlet所以在内存需要方面非常少，以至于没有产生full gc**

**通过jstat也能对上关系：**

****

1. 测试Tomcat在使用CMS下的吞吐率情况，和串行回收器和并行回收器在相同环境下做比较，给出几种回收器的报表截图，以及你对CMS的评价（从吞吐量等方面）。

由于使用的是servlet导致内存消耗比较小，所以servlet代码改造：

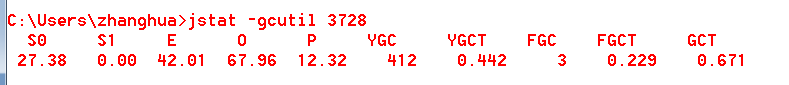


通过以上方式可以产生对象

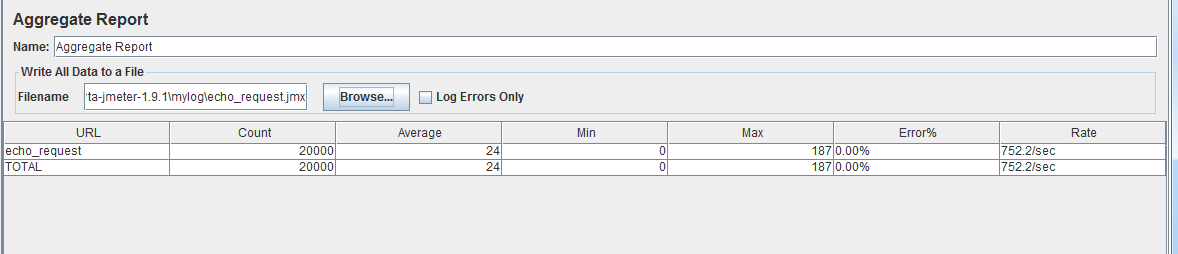
**2.1串行回收器参数：**

set CATALINA\_OPTS=-Xloggc:d:/gc.log -XX:+PrintGCDetails -Xmx40M -Xms4M -XX:+UseSerialGC -XX:PermSize=64M

jstat汇总结果：



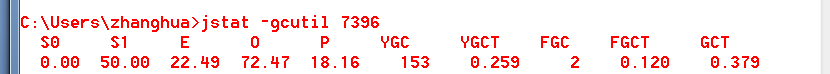
聚合报表结果：



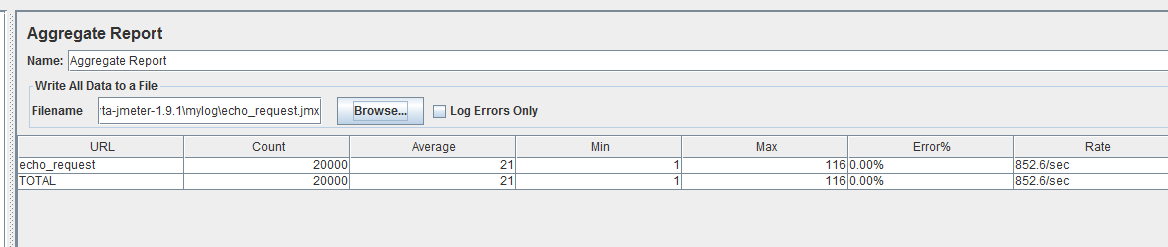
**2.2并行回收器：**

set CATALINA\_OPTS=-Xloggc:d:/gc.log -XX:+PrintGCDetails -Xmx40M -Xms4M -XX:+UseParallelGC -XX:+UseParallelOldGC -XX:ParallelGCThreads=2 -XX:PermSize=64M

jstat汇总结果：

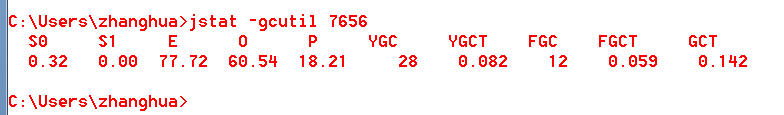


聚合报表结果

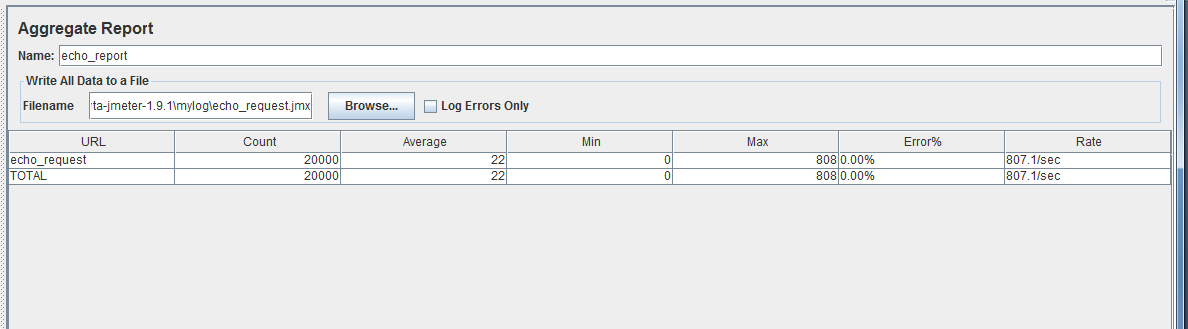


**2.3cms垃圾收集器：**

jstat汇总结果：



聚合报表结果



关于对cms评价：因为我是做nio方面的，对系统的响应时间和垃圾停顿时间比较在意，所以cms还是比较熟的，cms相对来说有以下特点：

1. 能保证系统的响应时间,减少垃圾收集时的停顿时间，对一些做监控的应用优其重要。
2. Cms采用Mark-Sweep会对堆产生空间碎片，但是可以指定在经过几次垃圾回收后再做总的空间整理
3. 从上面图中也可以看到，虽然fgc回收时间相对比并行收集器少，但总的回收次数确是多的，所以会导致系统停顿次数变多，导致吞吐量下降。