## 软件测试的流程

1、需求：阅读需求，理解需求，与客户、开发、架构多方交流，深入了解需求。--testing team

2、测试计划: 根据需求估算测试所需资源（人力、设备等）、所需时间、功能点划分、如何合理分配安排资源等。---testing leader or testing manager

3、用例设计：根据测试计划、任务分配、功能点划分，设计合理的测试用例。---testing leader, senior tester

4、执行测试：根据测试用例的详细步骤，执行测试用例。--every tester(主要是初级测试人员)

5、执行结果记录和bug记录：对每个case记录测试的结果，有bug的在测试管理工具中编写bug记录。--every tester(主要是初级测试人员)

6、defect tracking：追踪leader分配给你追踪的bug.直到 bug fixed。--every tester

7、测试报告：通过不断测试、追踪，直到被测软件达到测试需求要求，并没有重大bug.

8、用户体验、软件发布等。

## Windows/linux系统中测试环境搭建

2.1.先安装jdk,选择安装目录进行安装

2.2安装jre,直接安装

2.3.配环境变量,点击计算机右键属性,点击高级系统设置,找到javahome:jdk的安装目录;path:jdk下可执行文件的目录;classpath:java运行时需要查找class文件的目录.

2.4.打开cmd命令提示符窗口,输入java -version查看jdk的版本

2.5.安装tomcat,解压tomcat,在bin目录下找到startup.bat启动tomcat;点击shutdown.bat来关闭tomcat.

2.6.点击启动之后,在浏览器中输入http://localhost:8080是否能进入到tomcat页面

Linux系统搭建测试环境

先安装虚拟机,打开SSH 远程连接。选择quick connect，在弹出的页面里Host Name 为linux的ip（Linux中获取IP的方法是ifconfig）

User Name 为安装时设置的用户名 ,点击连接,输入密码,连接成功,在SSH File中新建java,tomcat文件夹,把压缩包拖进文件夹里.在SSH SHELL中首先安装jdk,先输入rpm -ivh j+tab来解压jdk,在根目录下,进入etc,把jdk的代码粘过去,然后输入java -version查

看jdk版本.安装tomcat,首先输入tar -xvf t+tab进行解压,输入mv对tomcat重命名,解压三次,改三次名,在tomcat里面的conf里修改端口号,在虚拟机上启动tomcat,浏览器中输入IP地址和端口号,看能否出现Tomcat的页面

## 3测试用例的要素，编写测试用例的方法以及如何保证测试用例的覆盖率与正确性

***测试用例要素：***

### PC端测试用例：

用例编号，用例类型，所属模块，用例目的，前置条件，具体操作步骤，预期结果，实际结果

APP测试用例：  
用例编号，用例类型，所属模块，用例目的，前置条件，具体操作步骤，预期结果，实际结果 测试版本 手机类型 手机系统

接口测试时用例：  
用例编号，用例类型，所属模块，用例目的，前置条件，具体操作步骤，预期结果，实际结果 请求数据 返回结果

### 性能测试用例：

用例编号，用例类型，所属模块，用例目的，前置条件，具体操作步骤 性能测试指标的来源 主要的性能指标 性能测试要点 预期结果，实际结果 请求数据 返回结果

**编写测试用例方法：**

等价类划分法，边界值分析法，错误推测法，功能法，异常测试法，因果图法，综合法

***如何保证测试用例的覆盖率与正确性：***

1、编写测试用例前，检查相关需求需求、设计文档是否有问题（功能描述不清，设计逻辑缺陷），如有问题找相关设计或者开发问清楚。

2、然后整理成需要覆盖的功能列表或者思维导图，功能列表包含新增和修改功能点，性能需求也要列出来（因为要整理对应的性能测试用例），同时还需要对既有功能进行一个梳理，检查是否会与其他功能有交互，整理出影响点。

3、把功能列表发给组员，并找时间进行会议评审，主要对功能等进行查漏补缺。

4、最后才行进测试用例编写，注意编写规范。

5、编写完毕后，把测试用例发给组员，开会进行评审，主要对检查点、用例规范进行查漏补缺。

6、执行测试用例过程中，发现用例不完善或者错误，需对测试用例进行及时的修改与调优

**7、测试完毕后对漏测的bug进行测试用例补充。**

## 4简述测试工作中接触过的数据库

1．无非就是oracle 存储量、吞吐量比较大，

但是查询速度慢，适合于不太及时响应并且数据运算量比较大的数据查询，

2．mysql 属于轻量级的 安装简单，开源免费 适合于简单数据结构的查询

3. Oracle是大型数据库而Mysql是中小型数据库。

4. Oracle支持大并发，大访问量

5. 安装所用的空间差别也是很大的，Mysql安装完后才152M而Oracle有3G左右，且使用的时候Oracle占用特别大的内存空间和其他机器性能。

6.**mysql**一般使用自动增长类型 **Oracle**没有自定增长类型

7.单引号处理**mysql**里可以可以用双引号包起字符串**Oracle**里只可以用单引号包起字符串

8.翻页的SQL语句处理，mysql处理翻页的sql语句比较简单，用limit开始位置，记录位置；Oracle处理翻页的sql语句比较繁琐。没个结果集只有一个rownum字段标明他的位置并且只能用rownum<100，不能用>，会提示sql语句错误。

9.长字符串的处理INSERT和UPDATE时最大可操作字符串长度小于等于4000个字节，如果要插入更长的字符串，请考虑用CLOB类型，方法借用oracle里自带的DBMS\_LOB程序包

10.空字符串处理：MYSQL的非空字段也有空的内容，ORACLE里定义了非空字段就不允许有空的内容

## 5.测试方案，测试计划文档中的内容

测试方案：

（1）文档说明

1. 编制说明
2. 适用范围

（2）概述

（3）测试环境

（4）测试流程

（5）测试结果

测试计划文档：

1. 引言：目的、背景、范围、定义、参考资料  
2. 测试内容：测试功能清单  
3. 测试规则：进入准则，暂停/退出准则、测试方法、测试手段、测试要点、测试工具  
4. 测试环境：硬件环境、软件环境、特定测试环境要求  
5. 项目任务：测试规划，测试设计，测试执行准备，测试执行，测试总结  
6. 实施计划：工作量估计、人员需求及安排、进度安排、其它资源需求及安排、可交付工件  
7. 风险管理

## 6.Python中string常用的方法有哪些

str.capitalize() 用法：将整个字符串的首字母变成大写，其余小写

str.upper() 用法：将字符串全部变成大写

str.isupper() 用法：检测字符串中所有的字母是否都为大写

str.lower() 用法：将字符串全部变成小写

str.islower() 用法：检测字符串中所有的字母是否都为小写

str.title() 用法：将字符串的每个单词的首字母大写，其余部分小写

str.istitle() 用法： 判断字符串是否首字母大写，且其他字母为小写（会忽视非字母字符）

str.count(sub[, start[, end]]) 用法：查找某字符串(sub)出现的次数，如果指定began（开始）和 end（结束）范围，则查找在该范围内出现子字符串的次数

str.index(sub[, start[, end]]) 用法：检测字符串string中是否包含子字符串 sub，如果存在，则返回sub在string中的第一个索引值(\*注意：空格算一个字符)，如果不存在就报错

## 7.Python中元组和列表的区别

Python的元组与列表类似，不同之处在于元组的元素不能修改。

元组使用小括号，列表使用方括号。

元组创建很简单，只需要在括号中添加元素，并使用逗号隔开即可

## 8.接口测试如何做的

我做接口的时候有postman，jmeter和自学的一点python。

我给您详细的说一下吧，我用postman但是做接口的，也是拿到接口文档,填上url路径和参数，可以选择get和post请求，返回相应的结果，我可以选择要查看的方式，可以选择html或者json形式来查看返回的结果。

jmeter也可以测接口，我们公司之前测接口的时候，拿到开发给的接口文档，然后创建线程组，选择配置元件，选择http 默认值，输入接口路径，输入接口需要传递的参数，选择get或post请求方式等等情况，选择返回json形式的结果，然后添加察看结果树，查看返回的结果是否正常。接口测试也可以设置断言匹配，这就是接口测试。

我也会一点python，我也自学了python 我曾经用python做过接口测试和一些简单的自动化测试我先说下用python怎么做接口测试的吧我当时用的是pycharm最新版，python是3.6来测的。在网上找了一些免费的接口,我把需要的包requests安装上导进去，安装就是pip install requests,导进去就是import requests.用requests.get放进去url地址和传参，然后复制给一个值，通过这值来获取响应报文和响应码，并打印，然后看返回结果请求值成功是200,错误返回404,500等，检查返回数据的正确性与格式是否正常，我一般都是返回json格式。我对python测接口测试目前就了解到了这么多。您还有什么想问的？

Postman和jmetr都是进行单个接口进行测试的方法。

Postman ：

在使用postman进行接口测试的时候，打开该软件，在该页面里填写路径、请求方式、请求参数（有get和post）所有的填写完成之后，点击“send”即可进行提交数据，会在“response”页面里展示结果，响应码和响应时间。response还可以以不同的方式查看，并且还可以看到cookies，headers信息。

1.使用自带的脚本对接口进行测试  
点击tests按钮，可以看到右侧有个snippets栏，里面就是postman内置的测试脚本，辅助对接口进行测试。选择其中的一个脚本，修改其中的参数，再次点击send，这时response的tests出现了一个1/1，说明执行一个测试并且通过测试，绿色pass说明验证通过。红色fail说明测试不通过，一个接口可以写多个脚本进行多次测试

会以json格式返回。

Jmeter：

用jmeter做接口测试时，首先拿到接口文档后先打开jmeter软件。

使用jmeter时，先解压一个jmeter压缩包，在jmeter解压后的文件夹里点击bin目录下的jmeter.bat进行启动jmeter，此时打开的黑窗口不能关闭，等待jmeter页面的开启。因为它是基于Java的，所以要配置环境变量。之后在测试计划上鼠标右键选择添加里的threads users里的线程组选项创建线程组。可以给该线程组起一个名字。设置循环次数。在这里选择永远的话会进入死循环。再在线程组上右键—添加—samples—HTTP请求。在该页面里进行填写请求方式、请求路径、请求参数、端口号等。在参数一栏里点击添加按钮，产生新的参数列表。在对应的位置上输入对应的值，之后在线程组上右键—监听器中添加图形结果、聚合报告、查看结果树三个监听器。对其进行分析。

测试完成之后我们会将bug以接口测试报告发给开发人员。接口测试报告里主要有bug的名称、bug的请求方式、请求参数、路径、响应码和响应报文以及编码格式。

## 简述python如何做接口测试、

我当时用的是pycharm。在网上找了一些免费的接口,我把需要的包requests安装上导进去，安装就是pip install requests,导进去就是import requests.用requests.get放进去url地址和传参，然后复制给一个值，通过这值来获取响应报文和响应码，并打印，然后看返回结果请求值成功是200,错误返回404,500等，检查返回数据的正确性与格式是否正常，我一般都是返回json格式。我对python测接口测试目前就了解到了这么多。您还有什么想问的？

## 简述通过python如何做自动化测试。

同样用的是pycharm，就拿之前做过的bbs项目来说，浏览器用的是火狐，首先在火狐上下载一个Xpath插件用来获取html页面标签，还需要下载一个火狐的驱动来实现访问功能。

然后安装selenium插件，安装方法是在控制台进入到python文件里script文件夹下输入pip install selenium。

安装完成后开始编写代码 新建一个python项目 输入from selenium import webdriver 来调入这个库 然后通过import time 调入时间这个库

然后使用webdrive.Firefox调用火狐浏览器，给他一个变量比如是“a”就用a.get（）括号里输入要访问的域名，然后让他跑起来看看能不能访问成功。成功了就进行下一步通过内置方法获取页面的标签（by\_name，by\_id，by\_xpath能用哪个用那个）

我主要用的是xpath，右击想要获的代码 选择xpath 然后选择copy 复制到括号里如果是点击就是.click,如果要填写参数就是.sendkeys（）括号里写要填写的参数。

如果有弹框就需要用d.switch\_to\_alert这个方法 给他赋个值比方是“b”，接着就比赛b.accept就是点击确认按钮。b.dismiss就是取消，a.quit（）就是直接退出浏览器，a.close仅退出当前窗口。

## 总结linux常用命令

1. 看日志：tail -f ../logs/catalina.out
2. 查看日志前100行：head -n 10 0 test.log
3. 查看日志后100行：tail -n 100  test.log
4. 关闭防火墙：service  iptables  stop
5. 开启防火墙：service  iptables  start
6. 查看防火墙状态：service  iptables  status
7. 解压两种方式：unzip  tar- xvf
8. 删除文件：rm -rf
9. 查看文件内容：cat

10. 授权：chmod + x \*.sh

11. 杀死进程：kill -9 进程号

12. 查看进程：ps -ef |grep 进程名

13. 重启虚拟机：reboot]

14. 查看当前使用的用户名：id <用户名>

15. 查看用户登录历史：last

16. 显示磁盘的空间使用情况以及挂载的分区列表：df -lh

17. 现实工作完整路径：pwd

18. 查看所有文件以及隐藏文件：ll

19. 创建文件夹：mkdir 20. 复制一个目录：cp

21. 注销：logout

22. 启动tomcat：./IMG_256 温馨提醒:双击图片可查看和编辑图片内容！ startup.sh

23. 关闭tomcat:  ./IMG_257shutdown.sh

24. 安装一个tomcat：rpm -ivh tomcat

25. 卸载一个jdk:  rpm -e--nodeps  jdk

26. 查看8080端口号有没有被占用：netstat -apn | grep 8080

27. 清空控制台：clear

28. 从日志中退出：ctrl +c

29. 强制保存并退出：wq！

30. 查看文件内容：cat

31. 显示系统日期：date -d

32. 编译：make

33. 查看IP地址：ifconfig

34. 查看磁盘：df

35. 下载命令：wget

36. 防火墙永久性生效关闭：chkconfig iptables off

37. 查看内存使用量和交换区使用量：free -m

38. 查看路由表：route -n

## 12.如何在linux搭建nginx+tomcat集群

在usr目录下创建java安装目录

cd /usr

mkdir java

2、将下载好的JDK文件通过ssh工具导入到/usr/java目

3、解压到当前目录

tar -zxvf jdk-8u60-linux-x64.tar.gz

得到文件夹 jdk1.8.0\_60

4、编辑配置文件，配置环境变更

/etc/profile

添加如下内容：

JAVA\_HOME=/usr/jdk

CLASSPATH=$JAVA\_HOME/lib/

PATH=$PATH:$JAVA\_HOME/bin

export PATH JAVA\_HOME CLASSPATH

5、source /etc/profile 应用一下

6、查看安装情况

java -version

在linux系统中的根目录下/usr目录中创建一个nginx文件夹；

在该文件夹下通过wget http://nginx.org/download/nginx-1.12.0.tar.gz命令在连接中下载nginx的安装包；

通过yum 命令安装GCC,ZLIB,PCRE,OPENSSL四个依赖环境

yum install gcc 安装gcc环境

yum install pcre-devel 安装pcre环境

yum install zlib zlib-devel 安装zlib环境

yum install openssl openssl-devel 安装openssl环境

通过tar -zxvf 命令安装后缀名为.gz的nginx安装包；

安装完成后在该目录下回出现一个nginx-1.xxx的文件夹；

通过cd命令进入该文件夹中，然后执行./configure命令，

初始化nginx配置

通过make命令执行make环境；

如果 Make 执行报错 执行yum -y install openssl openssl-devel

再次通过make install 安装make库；

通过whereis nginx命令查看linux环境下nginx的安装目录，

一般情况下，都是在/usr/local/nginx目录下

通过cd命令进入到nginx的安装目录下的sbin文件夹中

执行./nginx第一次没有显示任何信息（初始化启动）

需要再次执行./nginx 命令启动nginx，执行出现三行启动信息；

通过浏览器输入nginx所在的电脑的ip地址直接访问，

出现welcomne to nginx欢迎信息，则表示nginx安装并启动成功。

Nginx + tomcat 集群：

在linux系统中解压两个tomcat；

解压完成后，因为tomcat默认端口号是8080,所以需要将其

中一个tomcat的端口号改成别的；

     3.进入某一个tomcat的目录下的conf文件夹下，找到一个名为

server.xml的tomcat配置文件，在该文件中，将所有含有数字修改成其他值，避免两个tomcat端口号一样启动不了

     4.修改完成后，将两个tomcat启动，查看两个tomcat是否都能

成功启动并访问；

     5.当tomcat启动成功后，在进入到nginx的安装目录下，找到conf

文件夹，进入该文件夹后，找到名为nginx.conf的文件，通

Vi命令进入该文件，按i键进入文件编辑模式；

    6.在该文件的如下图位置进行如下图配置:



      7.当集群代码配置完毕后，继续要将nginx访问的反响代理信息

进行配置;

      8.继续在结束的server{}中，找到一个location{},在这个花括号中

的最后一行位置，添加如下代码:

 proxy\_pass http://集群名称;

     9.配置完毕后，通过:wq!命令保存并退出;

    10.启动nginx，启动完毕后，通过浏览器访问nginx所在的电脑的ip地址（相当于访问nginx）

     11.查看是否可以通过nginx访问到tomcat，如果能，则证明集群

成功。Nginx + tomcat 集群完毕。

## 简述性能测试，压力测试，并发测试，负载测试的区别

性能测试：软件测试的定义：模拟用户负载来测试系统在负载情况下，系统的响应时间，吞吐量等。（这里的负载指的是用户并发数）所以它的目的是为了获取系统的指标。

负载测试：软件测试的定义：在一定的软硬件环境上，通过不断的加大负载来确定在满足性能指标情况下所能够承受的最大用户数。所以它的目的是为了获取最大用户数。一般不超过80%cpu，正常情况工作下最大用户数数据。

压力测试，也叫强度测试。软件测试的定义：在一定的软件硬件环境下，通过高负载的手段来使服务器资源处于极限的状态，测试该系统在极限状态长时间运行是否稳定。包括系统指标，服务器性能指标。

综上所述：

一般情况下我们所说的性能测试就是在服务器指标不超过80%下的测试来获取性能指标，负载测试是测试的一个方法，通过不断调试并发数获取性能瓶颈。比如80个并发，这个叫80用户负载测试。通过80—>180这样的并发数变化过程，就叫做性能测试。也就是说，性能测试是通过不同的负载测试来实现的。

压力测试，就是高负载的情况下进行的，目的不是为了获取性能指标，而是想要了解系统是否稳定。这时候服务器的指标一般不超过90%。压力测试通过长时间的运行较性能测试更能容易发现内存泄露的问题。

简单来说，负载测试是个方法，性能测试是一个过程。压力测试是个高压力下的性能测试

## 简述如何通过jmeter如何做自动化测试

1.启动jmeter添加线程组。

2.为线程组添加元件和取样器

针对于http的简单应用，只需要添加HTTP Cookie 管理器，HTTP信息头管理器，HTTP请求默认值。

3.HTTP 请求默认值可以设置http请求的协议，域名，端口号。

4.三个元件添加好之后就可以添加HTTP 取样器了。

5.为线程添加监听器

监听器可以得到取样器中返回的结果，主要添加两种，聚合报告和察看结果树。

6.然后点击启动就可以了

## 简述如何通过jmeter如何做性能测试

先用badboy录制好脚本导入到jmeter中，比如登录功能，就在登录请求前设置同步定时器，设置好并发人数，当多个线程的用户到达该点后开始集结，达到集结人数后同时执行该功能，以获取并发情况下该功能的详细数据，对项目的性能进行评估推断

## 简述如何通过jmeter如何做并发测试

先用badboy录制好脚本导入到jmeter中，比如登录功能，就在登录请求前设置同步定时器，设置好并发人数，当多个线程的用户到达该点后开始集结，达到集结人数后同时执行该功能，以获取并发情况下该功能的详细数据，对项目的性能进行评估推断

## 17.简述jmeter断言，脚本关联的步骤

在需要添加断言的请求上添加后置处理器，在后置处理器中选择响应断言，在响应断言的文本框中添加断言的内容

## 简述jmeter如何做 性能分析

图形结果：

1. 样本数目: 所有的虚拟用户所操作的步骤
2. 最新样本: 最后一个虚拟用户操作的时间
3. 平均: 所有虚拟用户操作的平均时间
4. 偏离: 衡量服务器性能是否稳定的标准，偏离值越大，服务器性能越差，偏离值越小，服务器性能越好
5. 吞吐量: 服务器单位时间内对客户端响应数据对 的大小
6. 中值: 将所有虚拟用户所花费的时间进行排序，取出最中间的时间

查看结果树:

属性解释:

1.Text：调整服务器对客户端响应数据的查看格式

2.取样器结果: 包含内容有: response Code(响应状态码) response message (响应信息) DataEncoding (返回数据编码格式)

3.请求: 包含请求的url路径以及参数

4.响应数据: 服务器对客户端响应的数据（内容）

聚合报告:

属性解释:

1.Label: 请求的url路径

2.Samples：请求的虚拟用户数

3.Average： 平均响应时间

4.Median： 响应时间中值

5.90%Line： 百分之九十用户的响应时间

6.Min： 最小响应时间

7.Max: 最大响应时间

8.Error%： 错误请求占总请求数的百分比

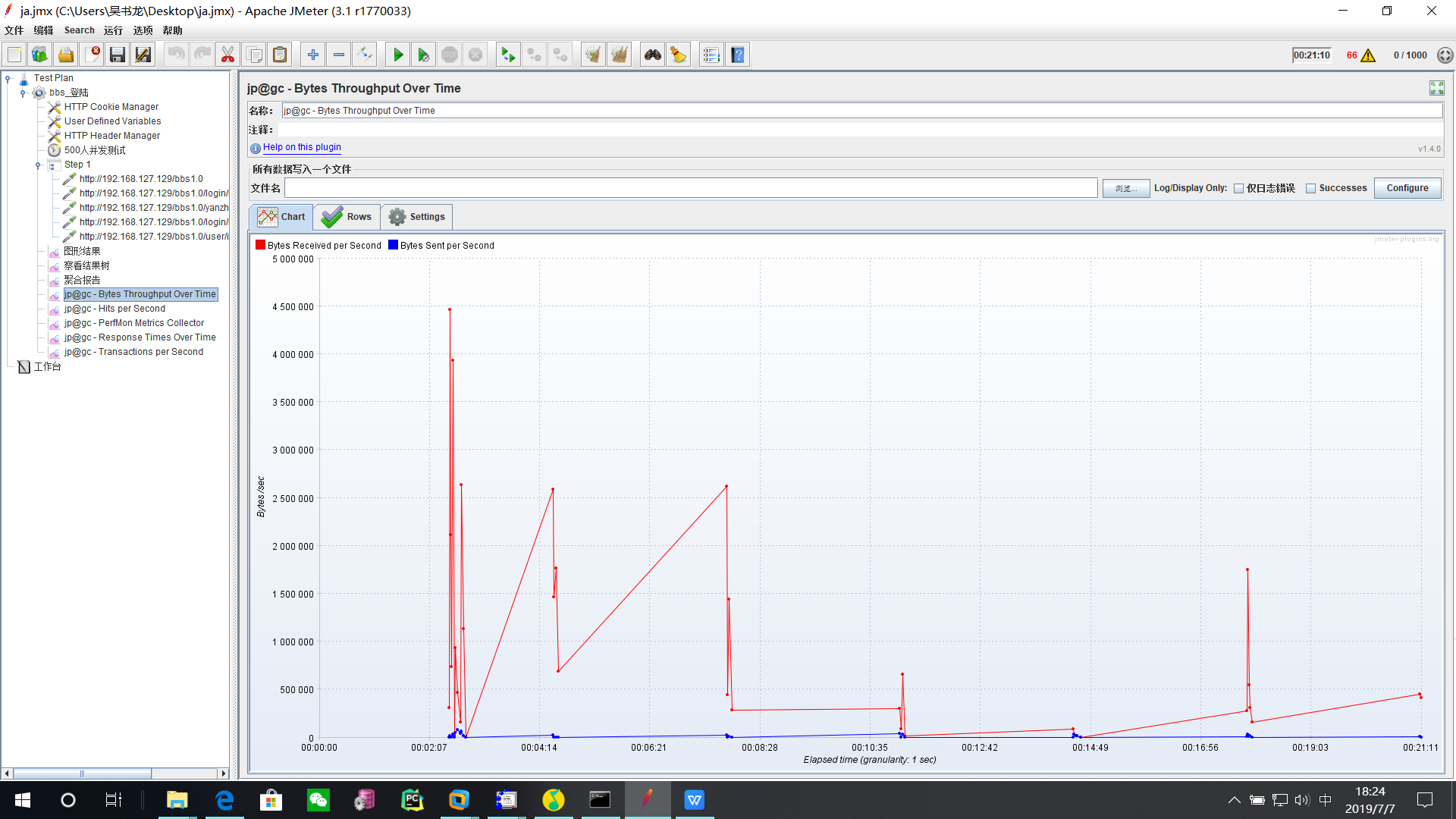
9.Throughput： 服务器每秒的吞吐量

10.Recevied： 接收数据的速率

Sent： 发送数据的速率

## Bytes Throughput Over Time（单位时间内服务器吞吐量变化趋势图）：

## 展示的是服务器发送和接收请求的个数。如图下所示：



图标解析：

1.横坐标表示的是时间轴

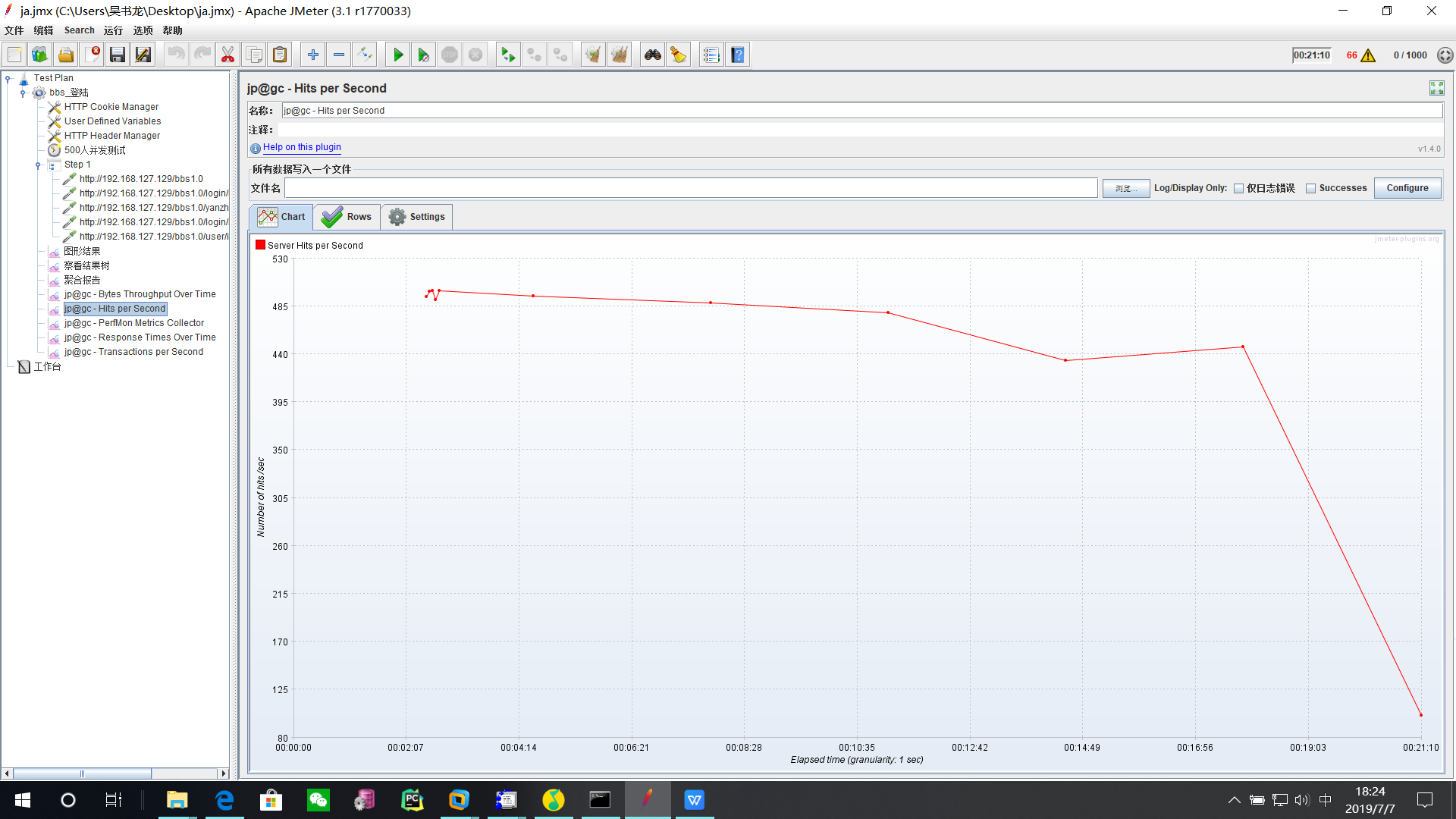
2.纵坐标表示的是服务器处理的数据量

3.红色的曲线表示的是接收的数据量

4.蓝色的曲线表示的是发送的数据量

## Hits per Second（每秒点击量）：

## 展示服务器每秒接收的请求的个数（客户端每秒向服务器端发送请求的个数）如下图所示：



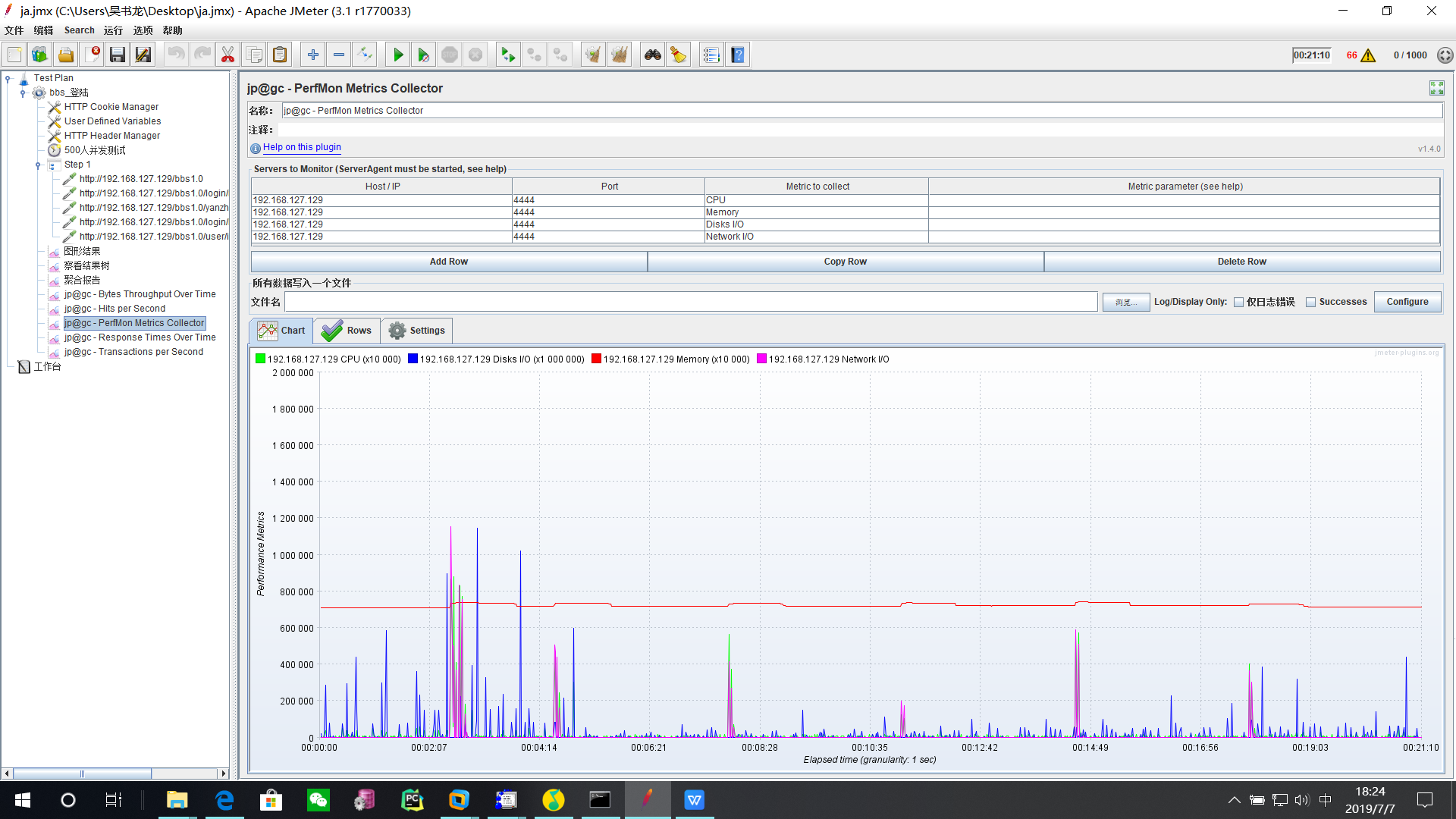
图表解析

1.横坐标表示时间轴

2.纵坐标表示请求的个数

## 2.6.Perfmon Metrics Collector（服务器硬件监听器）：

## 展示的是服务器硬件性能数据。如下图所示：

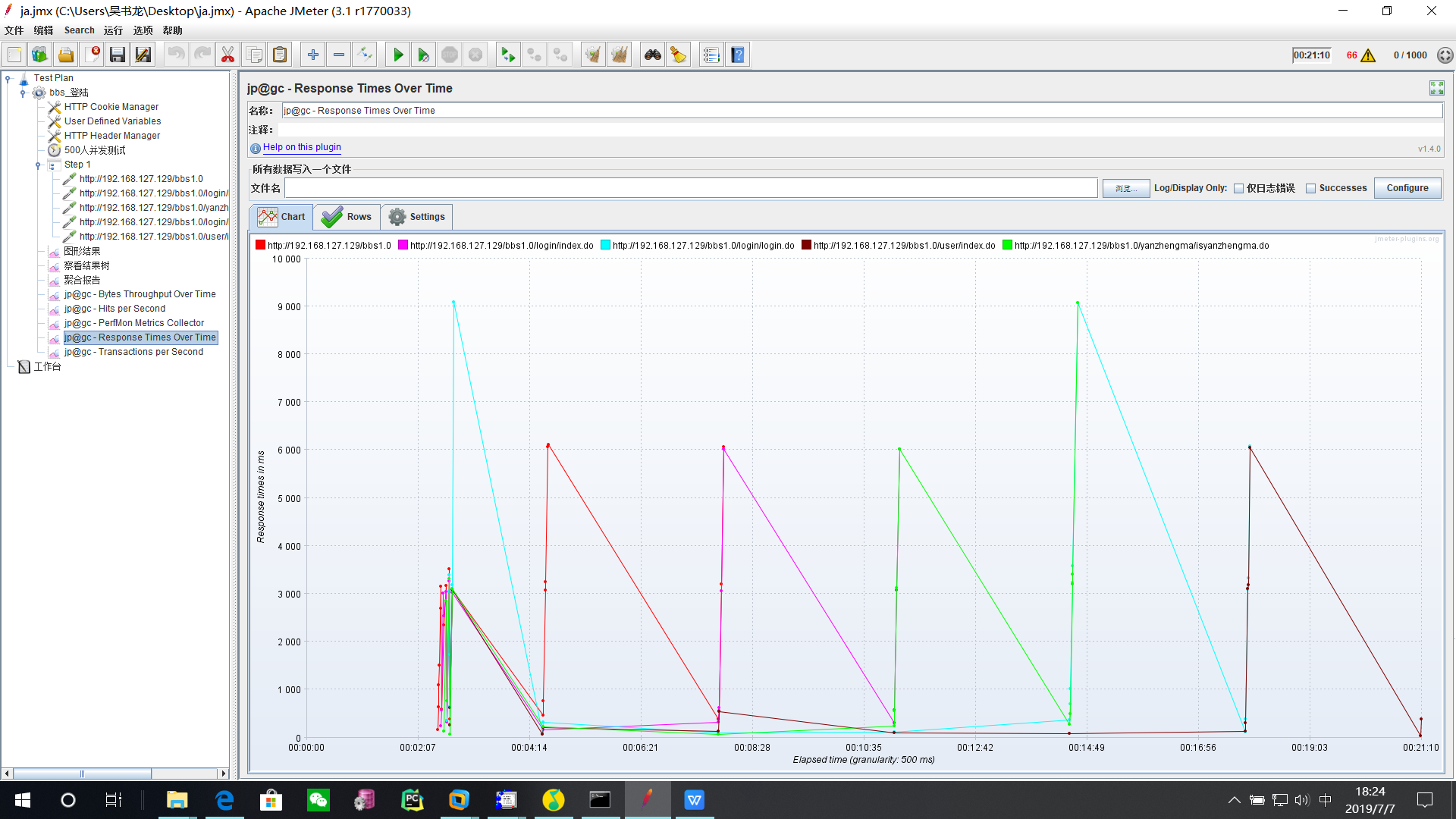


图标解析

1. Host/ip：填写服务器的IP地址
2. Port：填写serveragent插件的端口号（4444）
3. Metric collect：选择硬件类型
4. 红色的表示CPU变化的曲线图
5. 绿色的表示内存变化曲线图
6. 蓝色的表示硬盘的读写变化
7. 紫色的表示网络读写的变化曲线图

## 2.7.Response Times Over Time（单位时间内服务器响应时间变化曲线图）：

## 展示的是所有请求的响应时间变化趋势，如下图所示：



图标解析

1.登录功能的响应时间是蓝色的曲线图

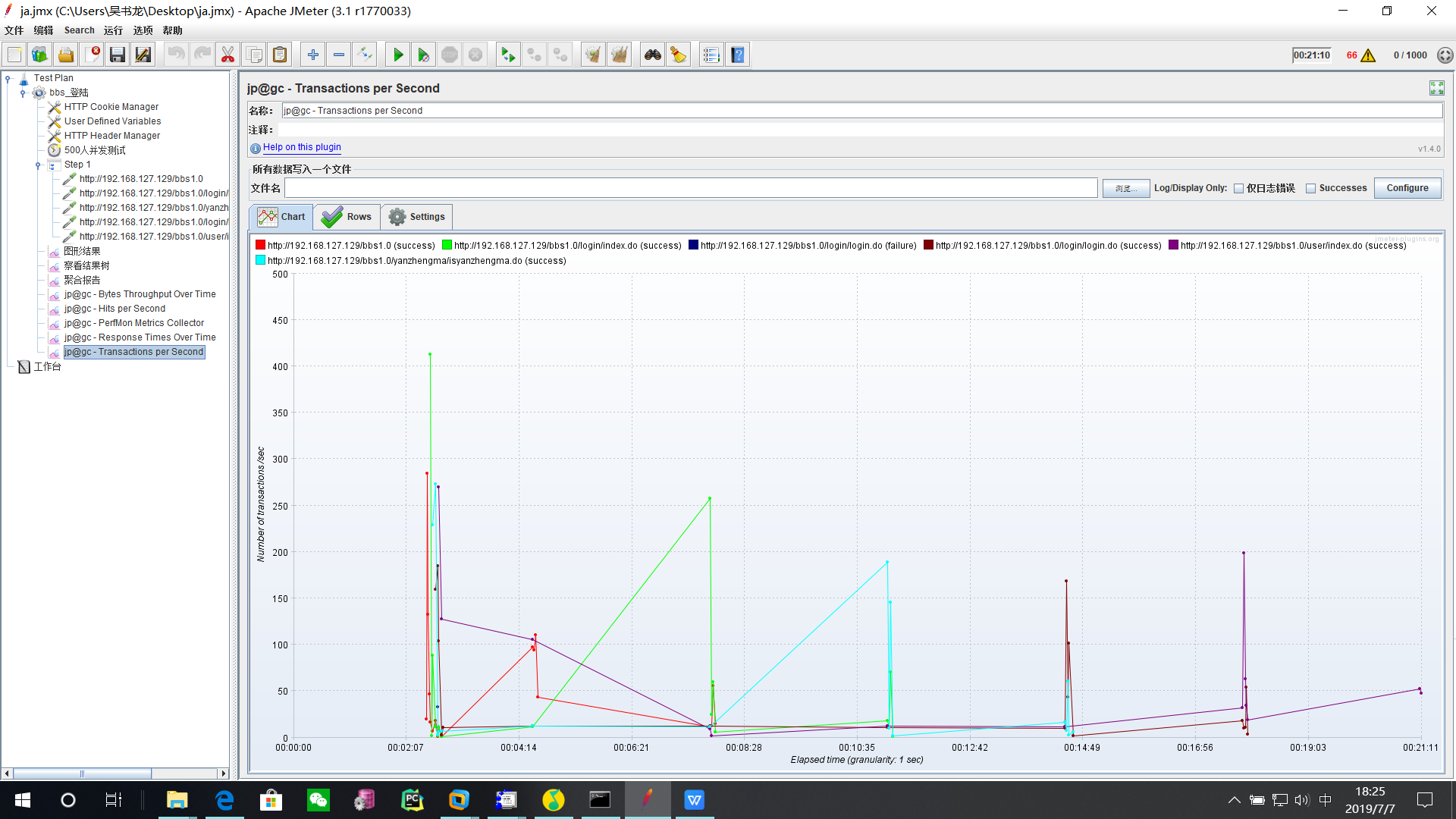
2.注册功能的响应时间是绿色的曲线图

3.横坐标表示的是时间轴

4.纵坐标表示的是服务器响应时间（单位：毫秒）

## 2.8.Transaction per Second（每秒事务数）：

## 展示的是服务器每秒处理的事务，如下图所示：



图标解析：

1.横坐标表示是时间轴

2.纵坐标表示的是每秒通过事务的个数

3.登录每秒通过的事物数是紫色的曲线

4.注册每秒通过的事务数是浅蓝色的曲线