**自我介绍**

经理，你好。我叫来禧玲，今年25岁，毕业于山西工商学院。2016年入职于北京科技

有限公司，在公司主要从事软件测试一职，主要是开发人员开发完成项目以后，我来进行测试，提出bug以及需要修改的一些地方，以文档的形式给测试经理或者用禅道进行提交bug。在这些年的工作经验中我所掌握的技能总结。主要有:

Html,js,数据库，测试方案，测试计划，测试用例，项目部署，python接口测试，自动化测试，linux系统，linux环境搭建，nginx+tomcat集群，postman接口测试，腾讯云，jmeter接口测试，自动化测试，性能测试，性能分析，压力测试。

1. html方面我了解的是一些常用的标签，from表单12元素，div，css和6种选择器，切割等
2. Js方面我了解的是Js常见的几种事件，鼠标单击事件，提交事件，失去焦点测试，内容改变事件，加载事件。
3. 数据库方面我接触的有oracle 和mysql我们公司一般使用的是mysql，它们在使用的过程中有很大的不同，比如字段类型mysql必须有主键，oracle可有可无，oracle的分页的关键词是rowman，mysql关键词是limit等等
4. Python方面我了解的是我会用python做接口自动化测试，接口测试多用于回归测试
5. 对于limit方面了解的是通过ssh远程连接工具，将本机和虚拟机连接在一起，通过linux命令系统进行操作，我们可以在linux系统下搭建测试环境变量配置jdk环境变量，解压tomcat，将项目部署在linux系统中，当一台tomcat不能满足请求时，我们需要解压多个tomcat，在进行nginx+tomcat进行集群
6. 对于postman，通常用来做接口测试，当开发人员传的接口不是以文档的形式而是只是部分接口时，我们可以用浏览器或postman进行测试，但浏览器返回的格式信息却不如postman方便，因需多使用postman接口测试，一般接口的请求方式有get和post，我们要根据具体的要求使用不同的传参方式
7. 对于腾讯云我们使用和linux系统差不多，首先我们需要申请服务器的ECS和数据库的mysql，然后再更改端口的配置，使用和linux同样的方式将项目部署在腾讯云上
8. Jmeter方面，通常做的是性能测试，当然也可以做接口测试，自动化测试压力测试性能测试在指定的测试基准中通过测试查看服务器性能是否满足要求压力测试在明确的测试基准情况下，通过不断访问的用户量，一直增加到服务器宕机或崩溃，从而测试出服务器的宕机和崩溃的临界值

html笔记：

HTML全称: hypertext markup language 超文本标记语言

后缀名的定义: 文件名称最后一个点后面的内容。

后缀名的作用: 决定当前文件的类型

常见的后缀名: txt 文本文档 jpg/jpeg/png 图片 gif 动态图片

doc/docx word文档 xls/xlsx excel表格 ppt/pptx 幻灯片文件

标签:

1. 定义: <>中+关键词

2. 分类: 单标签，双标签

标签解释:

head标签: 双标签 作用: CSS样式以及js代码存放的标签

title标签: 双标签 作用: 将标签中的内容显示在浏览器页面的页签上

body标签: 双标签 作用: 将标签中的内容显示在浏览器页面的浏览窗口中

center标签: 双标签 作用: 将标签中的内容水平居中

br标签: 单标签 作用: 换行标签

H标签: 双标签 作用: 将标签中的字体加粗，字体大小改变 ，H1 - H6 字体逐渐从大到小

font标签: 双标签 作用: 对标签中的字体改变样式

属性： size： 设置字体的大小，值得范围是: 1 - 7 ,字体从小到大

color：设置字体的颜色:

设置方式:

1. 通过颜色的英文单词设置颜色 color = "yellow"

2. 通过颜色的色值码设置颜色 color="#FFFF33"

font标签中通过size 设置字体大小时，有数值的限制，那么通过style属性设置时，就 不会有限制

格式:<标签 style = "样式属性: 样式属性值;"> <标签 style = "font-size:40px;">

img(图片标签): 单标签 作用: 将指定路径的图片显示在页面中

属性: src 指定显示图片的路径

路径: 1. 绝对路径: 具体盘符下的具体的某一个文件夹下的某一个 文件（D://aaa/bb.txt）

2. 相对路径: 以当前页面为参照路径，去指定另一个文件所 存在的位置（当前文件的路径(可省略)/另一个 文件的路径）

width 指定图片的宽度

宽度: 1. 通过具体的像素点 width="500px"

1. 通过占平路宽和高的百分比 width="70%"

**form表单元素**

**input 单标签 输入标签**

**属性: type 类型 决定当前输入标签的类型**

**name 名称 给当前标签起一个名称**

**1. 文本框 text 将用户在键盘上输入的字符以源码的形式显示出来的框**

**2. 密码框 password 将用户在键盘上输入的字符以隐码的形式显示出来的框**

**3. 单选框 radio 用户在多个选项中只能选择其中一个，并且单选按钮之间必须达到互 斥的效果**

**注意: 当单选按钮的name属性值保持一致时，单选按钮之间才能达到互斥的效果**

**4. 复选框 checkbox 用户可以在选项中选择多个**

**5. 文件框 file 通过文本框让用户可以选择文件进行上传**

**6. 下拉列表 select 标签 双标签**

**子标签 option 选项标签 双标签**

**7. 文本域 textarea 可以让用户进行多行输入**

**属性: rows 设置文本域的高度**

**cols 设置文本域的宽度**

**8. 按钮 button 可以点击的按钮**

**属性: value 将value属性值显示在按钮上**

**9. 提交按钮 submit 提交form表单**

**属性: value 通过value属性将默认的属性值覆盖**

**10.重置按钮 reset 将form表单中的数据清除掉**

**属性: value 通过value属性将默认的属性值覆盖**

**11. 隐藏域: hidden 此标签在页面中不显示**

**12. 图片 img/image**

form 表单

1. 定义: 页面数据收集器

2. 作用: 将页面中的数据进行封装打包

3. 属性: action 设置form表单数据发送的路径

method 设置form表单数据发送的方式

值: 1.get 轻量级的传输方式

优点: 传输速度快

缺点: 1. 传输量小，不能大于2KB;

2. 不安全（会将form表单中的数据显示在地址栏中）

2.post 重量级的传输方式

优点:1. 传输数据量大，不受限制；

2. 传输安全（不会将form表单中的数据显示在地址栏中）

缺点: 传输速度慢

CSS 的三种书写形式:

1. 标签内部样式: 将标签的样式设置在标签的内部

格式: <标签关键词 style="样式属性1:属性值1;样式属性2:属性值2;...样式属性n: 属性值n;">

2. 页面内部样式: 将标签的样式通过<style>标签设置在页面中

格式: <style>

CSS选择器{

样式属性1: 属性值1;

样式属性2: 属性值2;

样式属性3: 属性值3;

...

样式属性n: 属性值n；

}

</style>

3. 外部引入样式: 将CSS代码写到一个后缀名为.css的样式文件中，然后在页面中通过 link标签将样式文件引入

格式: <link href="css样式文件的路径" rel="stylesheet" type="text/css"/>

CSS的六种选择器:

1. id选择器: 通过标签的id属性值来获取当前标签(一次只能获取一个标签)

格式: #标签id属性值{

样式属性1: 属性值1;

样式属性2: 属性值2;

样式属性3: 属性值3;

...

样式属性n: 属性值n；

}

2. class选择器: 通过标签的class属性来获取当前标签（一次获取多个标签）

格式: .标签class属性值{

样式属性1: 属性值1;

样式属性2: 属性值2;

样式属性3: 属性值3;

...

样式属性n: 属性值n；

}

3. 标签选择器: 通过标签的关键词来获取标签（一次可以获取多个标签）

格式: 标签的关键词{

样式属性1: 属性值1;

样式属性2: 属性值2;

样式属性3: 属性值3;

...

样式属性n: 属性值n；

}

4. 通配符选择器: 将页面中所有的标签全部获取到

格式: \*{

样式属性1: 属性值1;

样式属性2: 属性值2;

样式属性3: 属性值3;

...

样式属性n: 属性值n；

}

5. 伪类选择器

1. 组合选择器

JS 全称: javascript java脚本代码

JS 作用: 提高页面和用户之间的交互

JS 格式:

<script>

function 自定义方法名称(){

方法体;

}

</script>

JS 调用: 通过触发JS中的事件来调用JS中的方法

JS 事件: 1. 提交事件 onsubmit (在点击form表单中的提交按钮后触发的事件)

2. 单击事件 onclick (在鼠标左键单击的时候触发的事件)

3. 鼠标离开事件 onblur (鼠标离开（失去焦点）的时候触发的事件)

4. 加载事件 onload (当页面在加载的时候触发的事件（优先于body标签 中的内容执行）)

数据库

1. 数据库的定义: 数据库只是一个服务，需要用可视化的工具才可以进行操作

2. 数据库的作用: 数据的存放地

3. 数据库的分类: 关系型数据库 和 非关系型数据库

4. 常用的数据库: 关系型（oracle ，SQL Server , MySQL） 非关系型（Redis）

oracle数据库

1. 定义: oracle是一个百万级的存储型数据库，安全级别较高（oracle数据库是有权限并且是按用户端的角色进行管理）

2. oracle可视化工具: PLSQL

3. oracle默认的超级管理员是: system 密码是: 装数据库时输入的口令

4. oracle用户操作的SQL语句:

创建用户: create user 用户名 identified by 密码

删除用户: drop user 用户名

授权用户: grant connect,resource,dba to 用户名

撤销授权: revoke connect,resource,dba from 用户名

5. oracle角色的分类: connect(连接角色) resource(资源角色) dba(数据库管理员)

6. oracle创建表的SQL语句:

create table 表名(

字段名 字段类型(大小) 是否为空(not null 不写默认就是可以为空) 是否为主键(primary key 不写默认就不是主键),

字段名 字段类型(大小) 是否为空(not null 不写默认就是可以为空) 是否为主键(primary key 不写默认就不是主键),

字段名 字段类型(大小) 是否为空(not null 不写默认就是可以为空) 是否为主键(primary key 不写默认就不是主键),

)

7. oracle数据库中表的主键: 给表中数据添加一个唯一标示 （主键在表中是具有唯一性并且不能为空）

8. 主键的作用: 表中既可以有主键，也可以没有主键，但是一般所有的表中都必须要有主键（主键可以保证每条数据的唯一性，以便于通过主键来快速查询众多数据中的某一条数据）

9. oracle数据库的事务: 操作数据库的过程被称之为数据库的事务，事务要么全部成功，要么全部失败，没有部分成功或者部分失败

10.事务的四大特性:

a. 原子性: 事务是一个整体，不可分割

b. 隔离性: 事务和事务之间不会相互影响

c. 一致性: 数据在保存到数据库之前和保存到数据库中是一致的

d. 持久性: 数据保存到数据库中后，除非删除，否则，永久都会在数据库中

11. oracle数据库的表的四个操作: 增，删，改，查

12. oracle表的四个操作中的增，删，改操作会对数据库表的数据有所改动，所以，这三种操作会产生事务

13. oracle数据库如果产生了事务，那么，在操作完成后，一定要将事务进行提交

14. oracle数据库事务的提交方式: 1. 手动点击事务提交按钮 2. 通过commit命令进行提交

15. oracle事务如果不提交，会造成表的死锁

16. oracle数据库的增删改查:

增: insert into 表名(字段1,字段2,字段3....字段n) values(值1,值2,值3...值n) （部分字段数据新增）

insert into 表名 values(表中所有字段的值，一定要按字段的顺序进行填写)

删: delete from 表名 （将指定的表名的表中数据全部删除，只是删除数据，表还在）

改: update 表名 set 字段1 = 值1,字段2 = 值2...字段n = 值n (将表中所有数据的该字段的值全部修改)

查: select 字段1,字段2,字段3...字段n from 表名

全字段查询: select \* from 表名 （将表中所有数据的字段全部查询出来）

部分字段查询: select 字段1,字段2,字段3...字段n from 表名 (将表中所有数据的指定字段查询出来)

**oracle和mysql的区别**

1. oracle的端口号是1521，Mysql的端口号是3306
2. Oracle是百万级存储量Mysql是十万级存储量
3. Oracle主键没有自增，Mysql主键有自增功能
4. Oracle表可以有主键,也可以没有，Mysql必须有主键
5. Oracle分页的关键词是Rownum，Mysql的关键字是limit
6. Oracle的安全性较高，通过权限来控制用户的访问内容，Mysql是一个账号，没有权限，所有Mysql的安全性低于oracle
7. Oracle有索引，Mysql没有索引
8. Oracle操作时会产生事务，并且事务需要手动提交，而Mysql没有事务，不需要提交事务
9. Oracle有全关联，Mysql没有全关联

**Eclipse下导项目的步骤**

1. 首先在eclipse中的项目列表里，通过imput导入项目，
2. 在eclipse中的server模块中添加Apache-Tomcat服务器
3. 当项目的javaresoure和jsp文件报错时，在项目名称上右键选择‘puildpath’然后在选择configure，puildpath，在弹出框中的libraries中将JDK和服务器指定，指定完毕后，点击apply按钮，将JDK和Tomcat中的Jar包引入项目中
4. 在servers中的apache-tomcat右键选择add and remove 在弹出框中双击右侧项目，项目会自动跳到右边
5. 修改数据连接信息，包含数据库连接IP地址，数据库名，用户名，密码
6. 在服务器上右键点击‘start’，启动服务器

**Linux环境下搭建测试环境**

1. 将后缀名为rpm的jdk安装文件通过ssh远程连接工具传送到虚拟机中
2. 在虚拟机中通过命令进入jdk安装包所在的目录下，用rpm-ivh jdk安装文件名称命令会对jdk进行安装，安装完成后在jdk的目录出现jdk1.8.2-121文件夹就是jdk的目录
3. 开始配置linux系统中的java环境变量
4. 通过命令进入/etc目录下，通过ls命令查看当前目录下是否存在一个orofile的文件
5. 通过vi profile进入profile文件
6. 按键盘上的i键进入编辑模式，将代码粘贴到done和unset i 中间，代码填写完毕后按Esc退出编辑模式，再按wq！（保存并退出），完事后按java-servion查看jdk安装信息能看到jdk版本信息则成功

**Tomcat**

1. 在/usc目录下创建一个文件夹（tomcat），然后把tomcat通过ssh远程连接工具传送到此文件夹下，
2. 在虚拟机中通过unzip对tomcat进行解压，然后通过chmod+x+.sh命令进行授权在输入./startuo.sh看看是否启动成功
3. 把项目的war包导入tomcat-webapps中启动，tomcat来自动生成项目文件

**Linux系统中安装NGINX步骤**

1. 进入到linux系统中/usr/中，通过mkdir命令创建一个nginx文件夹；
2. 在该目录下通过如下命令首先安装nginx在linux系统中所依赖的四个环境
   1. 通过yum install gcc命令安装GCC环境
   2. 通过 yum install pcre-devel命令安装PCRE环境
   3. 通过 yum install zlib zlib-devel命令安装ZLIB环境
   4. 通过 yum install openssl openssl-devel 命令安装OPENSSL环境
3. 安装完四个依赖环境后，通过如下命令下载nginx安装包:

wget http://nginx.org/download/nginx-1.13.7.tar.gz

1. 在nginx文件夹下通过ls命令可以看到一个后缀名为gz的安装包
2. 通过tar -zxvf 安装包进行nginx的解压
3. 然后在nginx文件夹中就可以看到一个nginx的文件夹
4. 通过cd命令进入该nginx文件夹，然后执行./configure命令安装nginx
5. 安装完成后通过make继续安装nginx
6. 执行完make后再执行make intsall命令安装nginx
7. 执行完make install命令后，通过whereis nginx查看nginx的安装目录
8. 通过cd命令进行入nginx的安装目录
9. 通过cd 命令进入nginx下的sbin目录
10. 执行./nginx初始化nginx
11. 再次执行./nginx开启nginx服务，知道看到三行启动信息为止
12. 通过浏览器输入127.0.0.1能看到欢迎语，则表示nginx安装并启动成功。

Nginx+tomcat集群

1. Nginx的安装
2. 在linux系统中的根目录下usr/目录下创建一个nginx文件夹
3. 在该文件夹下通过wget命令在连接中下载nginx的安装包
4. 通过yum命令安装gcc.pcre.zlip.openssl四个依赖的环境
5. 通过tar-zxcf命令安装后缀名为gz的安装包
6. 安装完成后在该目录下会出现一个ngin-1.xxx的文件夹
7. 通过cd命令进入该文件中，然后执行./configuer命令
8. 通过make命令执行make环境
9. 再次通过make instad安装make库
10. 通过whereis nginx命令查看nginx环境下nginx的安装目录，一般情况下都是在/user/local/nginx目录下
11. 通过cd命令进入到nginx的安装目录下的sbin文件夹中
12. 执行./nginx第一次（初始化）
13. 再次执行./nginx启动nginx，出现三行信息成功
14. 通过浏览器访问nginx所在电脑的ip地址直接访问，出现welcomme tonginx

则成功

jmeter的接口测试:

操作步骤:

1. 首先在jmeter的测试计划中添加一个线程组

2. 在线程组上鼠标右键线程组添加，在二级菜单中选择sampler

3. 在弹出的三级菜单中选择Http请求

4. 在Http请求中的服务器ip地址中写入服务器的ip地址，在端口号处填写服务器的端口号，

在url地址栏中输入要访问的接口路径

5. 当接口以get形式进行请求时，有两种方式进行传参，第一种，就是直接在url地址后面通过?追加参数

第二种，在url地址栏下的parameters中以key value的形式传递参数

6. 当接口以post形式请求时，则在parameters旁边的body data中以变量名1=值1&变量名2=值2&....

7. 接口参数中如果含有中文，则需要在url地址栏后面的content Encoding属性中设置参数的编码格式

8.如果对接口只是进行接口功能测试，只需要在线程组上添加监听器“查看结果树”，

在查看结果树中可以看到每一个虚拟用户请求接口后的结果

1. 性能测试

* 1. 定义: 性能测试就是在特定的测试场景中查看服务器是否满足要求， 通过模拟大量用户使用系统，查看系统的性能稳定性）

1.2 作用: 通过性能测试找寻出系统的性能缺陷，在系统正式交付使用之前，对系统的预期测试，避免出现当大量用户使用该系统，造成系统宕机或者崩溃

1.3 工具: apache-jmeter loadrunner 阿里云旗下的PTS

1.4 分类:

1.4.1 性能测试: 在指定好的测试基准中，通过测试，查看服务器性能是否满足服务器性能需求

1.4.2 压力测试: 在明确测试基准的情况下，通过不断增加访问的用户量，一直增加到服务器宕机或者崩溃，从而测试出服务器的宕机或者崩溃的临界值

1.4.3 负载测试: 在明确测试基准的情况下，保证服务器的各项性能指标都符合测试基准，通过不断增加用户量 得到的值，称之为负载测试，负载测试不会叫服务器测试宕机或者崩溃

1.4.4 强度测试: 在指定的测试场景中，以长时间对项目进行测试，查看服务器性能是否达标

# 一、测试场景

1、测试场景

2、虚拟线程数

3、启动时间

4、单位时间内线程启动数量

5、并发场景图

# 二、测试结果

1 、包含jmeter性能测试的8个监听器(图形结果,查看结果树,聚合报告,服务器吞吐量,硬件监听器,每秒事务数,每秒点击量,单位时间内响应时间变化图)

# 三、性能基准

1、cpu不能超过50%

2、内存不能超过70%

3、硬盘读写速率

4、网络传输速率

5、事务平均响应时间遵循2、5、8原则

# 四、性能分析

 1、图形结果:

1.1 首先通过图形结果,查看样本数目请求是否完毕

1.2 通过图形结果查看平均响应时间,是否超过2、5、8原则，所以该条通过，是否符合性能 测试基准

1.3 通过图形结果中的偏离值得出服务器的稳定性

1.4 给出服务器的每分钟的吞吐量

2 查看结果树

  2.1 通过查看结果树所有请求是否成功,是否有异常

3 聚合报告:

  3.1 通过图形结果得出偏离后服务器不稳定,再根据聚合报告的平均响应时间或90%用户的响应时间,来判定是否遵从2、5、8原则

  3.2 通过聚合报告查看最大和最小响应时间是否超出2、5、8原则，是否符合性能测试基准

  3.3 通过聚合报告error%得出容错率，错误率不能超过5%

4 服务器吞吐量变化曲线图:

  4.1 分析出吞吐量某个时间点上的峰值

  4.2 综合每秒点击量、每秒事务数、吞吐量得出变化曲线图得出，吞吐量是否会跟着每秒点击量、每秒事务数上升和下降

5 服务器硬件变化曲线图:

  5.1 首先查看出硬件监听器查看出cpu 内存 网络 硬盘读写 消耗最高的是什么

  5.2 通过测试场景,分析得出在集合点上,cpu 内存 网络 硬盘读写 是否发生响应变化

6 通过响应时间变化曲线图:

  6.1 分析出所有请求中消耗时间最大的的请求

  6.2 超出2、5、8原则,进行分析通过和不通过

7 每秒事务数变化图:

  7.1 分析得出在某个时间点的峰值

  7.2 综合每秒点击图得出,每秒事务数是否随着每秒点击上升还是下降

8 、 cpu有没有超过性能标准

9  、内存占有数/内存总数

# 五、性能测试BUG

1. 优先解决最致命的缺陷