哈尔滨工业大学

**计算机科学与技术学院/国家示范性软件学院**

**2019年春季学期**

**《软件架构与中间件》课程**

**实验报告**

**Lab 6：分布式系统质量测试**

|  |  |  |
| --- | --- | --- |
| **姓名** | **学号** | **联系方式** |
| 刘思琦 | 1163710228 | hit1163710228@163.com/18846183728 |
| 石昊 | 1163710226 | 2670041246@qq.com/15645077257 |

目 录

[1 实验概述 1](#_Toc12131469)

[1.1 实验目的 1](#_Toc12131470)

[1.2 实验要求 1](#_Toc12131471)

[2 实验内容与过程 1](#_Toc12131472)

[2.1 系统测试理论概述 1](#_Toc12131473)

[2.1.1 性能测试概述 1](#_Toc12131474)

[2.1.2 性能测试步骤 1](#_Toc12131475)

[2.1.3 性能测试指标 1](#_Toc12131476)

[2.1.4 压力测试 2](#_Toc12131477)

[2.2 Jmeter的安装配置 2](#_Toc12131478)

[2.3 具体测试过程 2](#_Toc12131479)

[2.4 【附】测试报告 3](#_Toc12131480)

[3 结对开发过程记录 7](#_Toc12131481)

[3.1 角色切换与任务分工 7](#_Toc12131482)

[4 实验总结 8](#_Toc12131483)

[文档全部完成之后，请在上述区域点击右键，选择“更新域”，在打开的对话框中选择“更新整个目录”]

# 实验概述

## 实验目的

1）学习针对分布式系统进行质量测试的方法、工具；

2）学习使用诸如Jmeter的系统质量测试工具，掌握其安装与配置的方法；

3）能够根据性能分析与评估报告对系统的架构细节进行调优或整体升级；

4）学习如何撰写测试报告。

## 实验要求

1）组队：2人结对成组

2）题目：分布式系统的质量测试

* 按照课程中7步骤测试法，对实验五的分布式进销存系统进行质量测试
* 应分析改造前后系统的质量指标
* 应包括常规性能指标
* 应关注压力测试
* 应给出测试报告

# 实验内容与过程

## 系统测试理论概述

### 2.1.1 性能测试概述

性能测试主要是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试。性能测试主要分为负载测试、压力测试、容量测试、网络性能测试等等。

### 2.1.2 性能测试步骤

性能测试分为七个步骤：制定目标和分析系统，选择测试度量的方法，选择相关技术和工具，制定评估标准，设计测试用例，运行测试用例，分析测试结果。

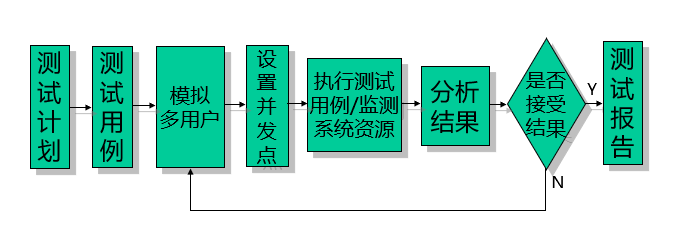
### 2.1.3 性能测试指标

主要的性能测试指标有：响应时间、内存、磁盘、处理器、网络。

### 2.1.4 压力测试

压力指的是在同一时间或某一时间内，向系统发送预期数量的交易请求（并发交易请求、递增交易请求、并发递增交易请求）。压力测试指测试系统在不同压力情况下的效率状况，以及系统可以承受的压力情况。

压力测试的流程如下图所示：

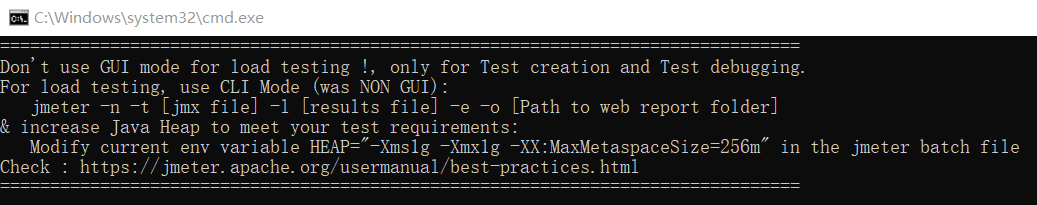


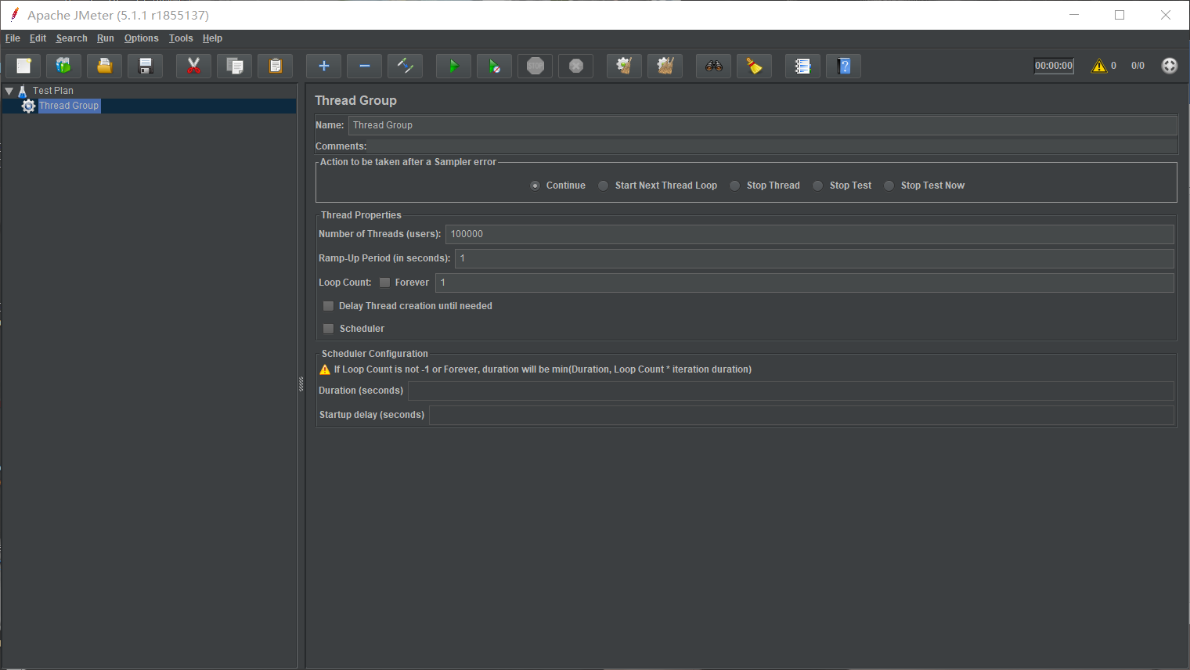
## Jmeter的安装配置

首先去官网下载Jmeter，下载地址为<http://jmeter.apache.org/download_jmeter.cgi>。下载

完成之后解压zip包。打开解压目录下的bin文件夹，双击运行jmeter.bat文件即可启动Jmeter。

启动后的界面如下图所示：





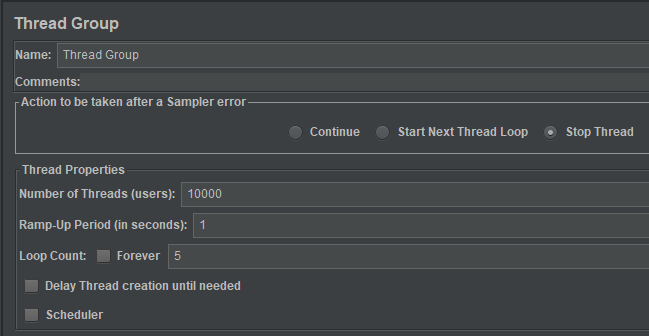
## 具体测试过程

第一步：启动Intellij IDEA，将待测试的web系统部署在Tomcat服务器上，此时，在

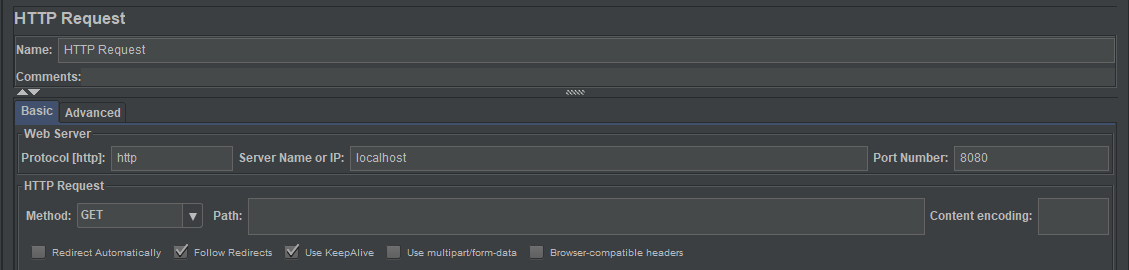
浏览器中输入localhost:8080或者输入ip地址:8080即可访问到web系统。

第二步：按照2.2中叙述的方法启动Apache JMeter系统质量测试工具。

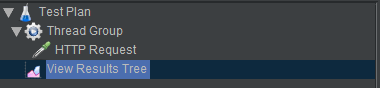
第三步：File → New → Test Plan，然后右键Test Plan → Add → Threads (Users) → Thread Group，设置循环的次数、并发的线程数量和全部线程的释放时间。首先设置为循环1次，10000个并发线程，全部线程在1s内释放。



第四步：右键Thread Group → Add → Sampler → Http Request来创建要测试的http请求的url信息。添加http请求相关信息和参数



第五步：添加查看结果的方式。右键Test Plan → Add → Listener → View Results Tree/汇总报告，此处选择的是查看结果树和查看汇总报告的方式。



第六步：点击绿色三角形形状的start按钮，开始进行测试。

## 【附】测试报告

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **学生信息管理系统性能测试报告**  **1. 引言**   * 1. **编写目的**   编写该系统性能测试报告主要有以下几个目的：   1. 通过对测试结果的分析，得出对软件系统质量的评价 2. 分析测试的过程、产品、资源、信息，为以后制定测试计划提供参考 3. 评估测试执行和测试计划是否符合 4. 分析系统存在的缺陷，为修复和预防脆弱性（vulnerability）提供建议    1. **背景**   原系统是基于mysql + servlet + jsp实现的一个Java web学生信息管理系统。后端采用的是dao – model – servlet三层架构，前端由Jsp + JavaScript整合EasyUI框架实现了友好的用户界面。在实验五中，已经通过数据层、计算层和表示层的三个层面上的改造，将原有的遗留Web系统改造为满足高性能、高可用、高可靠等质量属性的复杂软件系统。在数据层上应用React技术来重新设计Web系统的前端架构，并重写前端，使其具有React优秀的性能。在计算层上使用Nginx负载均衡技术实现基于流量分发的完全分布式系统集群的负载均衡架构，同时使用Hadoop大数据计算技术实现海量用户访问请求的log日志统计。在数据层上从系统的核心需求出发，重新设计数据库结构，对数据库结构在表层面上进行优化，所应用到的技术与思想包括Sharding-JDBC数据库分库分表技术，数据库关系模式范式理论以及数据库设计的ACID原则。同时还计划实现基于Redis的系统缓存架构，即所有的查询结果都放进了缓存，也就是把MySQL查询的结果放到了Redis中去，然后第二次发起该条查询时就可以从Redis中去读取查询的结果，从而不与MySQL交互，达到优化的效果，Redis的查询速度之于MySQL的查询速度相当于内存读写速度之于硬盘读写速度。     * 1. **用户群**   此系统主要面向大学教务处的信息管理员，辅导员，以及其他和学生信息的  增删改查有关的人员。   * 1. **测试对象**   测试对象为此系统的各方面性能水平。   * 1. **测试阶段**   系统集成测试。   * 1. **测试工具**   Apache Jmeter.   * 1. **参考资料**  1. 课程PPT——软件架构测试方法 2. <https://www.cnblogs.com/0201zcr/p/5046193.html> 3. <https://blog.csdn.net/u012111923/article/details/80705141>   **2. 测试概要**   * 1. **测试执行**   此次测试严格按照项目计划和测试计划执行，按时完成了测试计划规定的测  试对象的测试。针对测试计划规定的测试策略，在测试执行中都有体现，在测试的执行过程中，依据测试计划和测试用例，对系统进行了完整的测试。   * 1. **测试用例**  1. 功能性   系统实现的主要功能，包括各项数据的查询、添加、修改、删除；  系统实现的次要功能，如模糊查询；  需求规定的输入输出字段，以及需求规定的输入限制。   1. 易用性   操作按钮提示信息的正确性、一致性、可理解性；  限制条件提示信息的正确性、一致性、可理解性；  必填项标识；  输入方式可理解性；  中文界面下数据语言与界面语言的一致性。   1. 正确性   负载均衡分发的正确性；  插入数据分库分表的正确性。  **3. 测试环境**  **3.1 主机环境**   |  |  | | --- | --- | | 操作系统 | Windows 10 家庭中文版 | | 处理器 | Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99GHz | | 内存（RAM） | 16.0GB | | 服务器 | apache-tomcat-8.5.42 | | 数据库 | mysql-5.7.26-winx64 | | JDK版本 | jdk1.8.0\_211 | | IDE | IntelliJ IDEA 2019.1.3 |   **3.2 集群环境**   |  |  |  |  | | --- | --- | --- | --- | | 服务器 | 内存 | 容量 | 功能 | | Ubuntu 18.04 | 4GB | 60GB | 负载均衡器 | | Ubuntu 18.04 | 3GB | 40GB | 负载节点 | | Ubuntu 18.04 | 3GB | 40GB | 负载节点 |   **3.3 网络拓扑**  C:\Users\湖人总冠军\AppData\Roaming\Tencent\Users\1031901796\TIM\WinTemp\RichOle\[MPQFAB%KL20{[[M@QR4BDL.png  **4. 测试结果**  **4.1 常规性能指标测试结果**  经过向系统中插入各种测试数据，并对数据进行修改，插入错误输入，检查数据的分库分表情况，可以确定此系统满足常规的性能指标，如功能性，易用性，正确性等。  **4.2 压力测试结果**  **第一组测试：**线程数10000，Ramp-Up Time1s，循环次数1  测试结果：  C:\Users\湖人总冠军\AppData\Roaming\Tencent\Users\1031901796\TIM\WinTemp\RichOle\WP~0W6NKGOSUT{8{G2_Y0GH.png  结果分析：从结果中可见，异常率为0，吞吐量为2801.9/sec，表明系统完全可以承载并发量10000的访问量。  **第二组测试：**线程数10000，Ramp-Up Time1s，循环次数5  测试结果：  C:\Users\湖人总冠军\AppData\Roaming\Tencent\Users\1031901796\TIM\WinTemp\RichOle\E7ARS@0F@W{(1FCXBL)8VZ7.png  结果分析：从结果中可见，在这种情况下，异常率有了大幅度的上升，说明在这种压力测试下，系统的性能有了很大的下降。  **第三组测试：**线程数10000，Ramp-Up Time1s，循环次数3  测试结果：  C:\Users\湖人总冠军\AppData\Roaming\Tencent\Users\1031901796\TIM\WinTemp\RichOle\2M`ZF{)_)_SF68OO3DK8Y6G.png  结果分析：从结果中可见，在这种情况下，相比于第二组，异常率有所下降，吞吐量也有上升，但是异常率依然较高，说明在循环次数为3的压力测试下，系统的性能也有较大幅度的下降。  **第四组测试：**线程数100000，Ramp-Up Time1s，循环次数1  测试结果：电脑死机了，系统报错：  C:\Users\湖人总冠军\AppData\Roaming\Tencent\Users\1031901796\TIM\WinTemp\RichOle\C2QUWTDTW9[QNGO_(]@@N15.png  结果分析：运行内存超出JVM初始化堆内存。。  **5. 测试结论**  经测试，该系统在并发量达到20000-30000时性能就开始产生了下降，系统的性能瓶颈主要在于tomcat服务器的并发量以及JVM虚拟机的堆内存空间。采用负载均衡机制后，负载服务器越多，分布式系统可承载的并发量也会越高。 |

# 结对开发过程记录

## 角色切换与任务分工

表1-1结对开发角色与任务分工

| 日期 | 时间(HH:MM - HH:MM) | 驾驶员角色 | 领航员角色 | 本段时间的任务 |
| --- | --- | --- | --- | --- |
| 2019/6/22 | 18:00-19:00 | 刘思琦 | 石昊 | 配置环境，进行测试 |
| 2019/6/22 | 19:00-21:00 | 刘思琦 | 石昊 | 撰写报告 |

# 实验总结

经过本次实验，把课堂上学习到的测试思想及流程进行了一次实践，通过实践更深刻的体会到了测试的思路以及流程，同时经过压力测试，对系统可承载的并发量也有了一个较为清晰的认知。