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**

**2020年春季学期**

**《软件架构与中间件》课程**

**实验报告**

**Lab 6：系统质量测试**

|  |  |  |
| --- | --- | --- |
| **姓名** | **学号** | **联系方式** |
| 黄项宇轩 | 1173710106 | 1171845717@qq.com |
| 任永鹏 | 1173710107 | 805402160@qq.com./18800420869 |

目 录

[1 实验概述 1](#_Toc40518527)

[1.1 实验目的 1](#_Toc40518528)

[1.2 实验要求 1](#_Toc40518529)

[2 实验内容与过程 1](#_Toc40518530)

[3 结对开发过程记录 12](#_Toc40518531)

[4 实验总结 15](#_Toc40518532)

[5 教师评语 15](#_Toc40518533)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号：** | **1173710106** | **姓名：** | **黄项宇轩** | |
| **学号：** | **1173710107** | **姓名** | **任永鹏** | |
| 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. 2人结对成组 2. 按照课程中7步骤测试法，对实验五的系统进行质量测试。 3. 分析改造前后系统的质量指标(包括常规性能指标) 4. 关注压力测试 5. 给出测试报告 | | | | |
| 实验内容与过程 | | | | |
| 测试一：Web程序jmeter测试：  首先进行badboy录制    然后导出jmx文件    导入jmeter    实现参数化：    进行指标参数监控：    测试结果：  200并发测试：  400并发测试：    800并发测试：    最后统计结果：      下面为上面表格数据的解释：  #Samples：样本数量。多少个请求  Average：平均响应时间（单位：）。默认是单个Request的平均响应时间，当使用了TransactionController时，也可以以Transaction为单位显示平均响应时间  Median：中位数，也就是50%用户的响应时间  90%Line：90%用户的响应时间  95%Line：95%用户的响应时间  99%Line：99%用户的响应时间  Min：最小响应时间  Max：最大响应时间  Error%：本次测试中出现错误的请求的数量/请求的总数  Throughput：吞吐量。默认情况下标示每秒完成的请求数（具体单位如下图）  KB/sec：每秒从服务器端接收到的数据量  总结：  从上面的统计中可以看到，并发数越多，平均延迟越高，期望的99%的用户延迟得到明显增加，吞吐率变得越来越低。  最后我测试了一下极端情况的软件的性能表现：  设置并发数为2w      这种情况下，软件的平均延迟显著增加，错误率在10%左右  测试二：负载均衡压力测试  压力测试使用apache bench进行压力测试  命令行为ab -c 10 -n 100 + 测试url  意思为模拟10个用户共发送1000个请求  详细测试数据如下：  服务器软件:nginx / 1.10.3  服务器主机名:192.168.174.129  服务器端口:80  文件路径:/ # /登录? = % 2 findex重定向  文档长度:800字节  并发级别:10  测试时间:1.325秒  完成要求:100  失败的请求:0  Non-2xx反应:100  传输总量:98600字节  传输的HTML: 80000字节  每秒请求数:75.44[#/秒](平均)  每次请求所需时间:132.550 [ms](平均值)  每个请求的时间:13.255 [ms](所有并发请求的平均时间)  传输速率:72.64 [Kbytes/sec]接收  连接次数(Miss)  最小均值[+/-sd]中值最大值  连接:0 0 0.2 0 1  处理:1 111 165.0 23 821  等待:1 111 165.0 23 821  总数:1 111 165.1 23 821  在一定时间内处理的申请的百分比(毫秒)  50% 23  66% 114  75% 172  80% 255  90% 325  95% 350  98% 760  99% 821  100% 821(最长请求)  下面是截图    为比较负载均衡的效果，测试不进行负载均衡的情况下的数据  控制变量同样为10个用户共发送1000个请求  测试数据如下：  服务器软件:  服务器主机名:localhost  服务器端口:8081  文件路径:/ # /  文档长度:800字节  并发级别:10  测试时间:2.528秒  完成要求:100  失败的请求:0  Non-2xx反应:100  传输总量:95500字节  传输的HTML: 80000字节  每秒请求数:39.55[#/秒](平均)  每次请求所需时间:252.833 [ms](平均值)  每个请求的时间:25.283 [ms](所有并发请求的平均时间)  传输速率:36.89 [Kbytes/sec]接收  连接次数(Miss)  最小均值[+/-sd]中值最大值  连接:0 1 4.9 0 35  处理:0 247 603.4 15 2195  等待:0 222 602.2 6 2147  总计:0 248 605.5 15 2195  在一定时间内处理的申请的百分比(毫秒)  50% 15  66% 54  75% 87  80% 145  90% 2014  95% 2015  98% 2051  99% 2195  100% 2195(最长请求)  下面为截图    下面是关键数据统计：    根据对比测试数据可以发现，使用负载均衡，测试时间，请求时间明显缩短。  **数据库测试**  **制定目标和分析系统**  作为一个分布式进销存系统后台数据库，客户必然希望能够在自己各家分店，各台终端稳定流畅地使用本系统，这意味着对于我们来说，系统需求低延迟短响应时间和高并发数。  系统后台数据库使用关系型数据库Mysql，并使用Mycat对数据库进行分库分表以减轻单一数据库压力，提高读写效率。  **选择测试度量的方法**  遗留系统开发原成员在本次实验中并不齐全，相关流程、角色、职责分配从简，由两名组员共同承担。  **选择相关技术和工具**  在之前的改造过程中，曾使用Apache Benchmark进行测试，ab是apache自带的压力测试工具。ab非常实用，它不仅可以对apache服务器进行网站访问压力测试，也可以对或其它类型的服务器进行压力测试，比如nginx、tomcat、IIS等。本次实验应指导书推荐，最终选择jmeter做本次测试的性能测试工具。Apache jmeter可以用于对静态的和动态的资源的性能进行测试。它可以用于对服务器、网络或对象模拟繁重的负载来测试它们的强度或分析不同压力类型下的整体性能。可以使用它做性能的图形分析或在大并发负载测试服务器、脚本、对象。  **制定评估标准**  缺少大量过去的，扩展的或者将来可能发生的数据，因此定义最初的测试作为标准，利用它和所有后来进行的测 试结果进行对比。  **设计测试用例**  对仓库表warehouse进行测试，不使用Mycat和使用Mycat采用相同的请求语句以控制无关变量，请求地址位于黑龙江的所有仓库的地址    设置测试情景，遇到错误样本继续测试    设置线程数，即用户Number of Threads(users)    告知JMeter要在多长时间内建立全部的线程Ram-Up Period(in seconds)    设置循环次数Loop Count    每轮循环使用相同用户    **运行测试用例**    不使用Mycat    使用Mycat      不使用Mycat    使用Mycat      不使用Mycat    使用Mycat    **分析测试结果**  可以看到随着请求数和并发数的增大，耗时和单次延迟都变得更大了，在效率方面Mycat的优势并不特别明显，但是如果观察出现错误的概率就能发现，随着并发数上升，数据库开始不堪重负，错误率越来越高，但是使用了Mycat进行分库分表，缓解了单一数据库压力，在错误率上明显低于单一数据库。 | | | | |
| 结对开发过程记录 | | | |  |
| **（1）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 | | --- | --- | --- | --- | --- | | 2020.6.16 | 9:00-10:00 | 黄项宇轩 | 任永鹏 | 熟悉实验内容，进行任务分工 | | 2020.6.16 | 10:00-11:46 | 任永鹏 | 黄项宇轩 | 进行web程序jmeter测试 | | 2020.6.16 | 13:00-15:00 | 黄轩宇轩 | 任永鹏 | 进行数据库mysql-jmeter测试 | | 2020.6.16 | 15:00-15:50 | 任永鹏 | 黄项宇轩 | 进行负载均衡压力测试 | | 2020.6.16 | 15:50-16:30 | 黄项宇轩 | 任永鹏 | 撰写报告 | | | | | |
| **（2）工作日志** | | | | |
|  | | | | |
| 表1-2 结对开发工作日志 | | | | |
| | 日期/时间 | 问题描述 | 最终解决方法 | 交流过程 | | --- | --- | --- | --- | | 2020.6.16 | http请求返回405 (Method Not Allowed) | 由post请求改为get请求，请求静态资源时用get请求； | Csdn查找资料 | | 2020.6.16 | 发送HTTP请求返回415状态码的解决办法 | 在请求头加入"Content-Type":"application/json" | Csdn查找资料 | | 2020.6.16 | java.sql.SQLException: The server time zone value '�й���׼ʱ��' is unrecognized | Mysql版本过高时会出现的错误，在数据库连接url中添加?serverTimezone=UTC |  | | 2020.6.16 | Variable Name must not be empty for element:JDBC Connection Configuration | Variable Name：配置元件的的所有配置所保存的变量，自定义变量名称，多个jdbc配置元件可以使用相同的变量，但是只有一个会被jdbc请求使用，这里定义的变量名称会被之后的jdbc请求引用，以此来判断请求使用的是哪个配置元件所定义的数据库配置，必填项 |  | | | | | |
| **（3）结对开发工作现场照片** | | | | |
| 请其他同学帮助拍摄结对开发现场照片至少2张。 | | | | |
|  | | | | |
| 实验总结 | | | |  |
| 之前实验压力测试一直使用的是apache的ab进行测试，本次使用的软件是jmeter，并且通过badboy录制脚本操作，简化了jmeter的使用，方便了压力测试的参数化，检查点等方面的操作。实验进行到这一步，一整个学期的内容终于组成了一个完整的闭包，从对遗留系统一步步的改造，数据层负载均衡，计算层缓存与分库分表，表示层性能优化，最后使用专业的性能测试工具对这个相较课程进行前已经大大改进的分布式系统做性能测试，压力测试。虽然只能算是体验式的，但是却足够成为我们在这一方向的第一步，往往是最难的第一步，之后的学习工作中，有了这个基础就能进一步练习相关技能，课程无疑是提高技术水平重要的敲门砖。 | | | | |
| 教师评语 | | | |  |
|  | | | | |