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 5：分布式系统整合实验**

|  |  |  |
| --- | --- | --- |
| **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](#_Toc41764781)

[1.1 实验目的 1](#_Toc41764782)

[1.2 实验要求 1](#_Toc41764783)

[2 实验内容与过程 1](#_Toc41764784)

[3 结对开发过程记录 23](#_Toc41764785)

[4 实验总结 25](#_Toc41764786)

[5 教师评语 25](#_Toc41764787)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. 尝试综合运用计算层、数据层和表示层的架构技术 2. 实现满足高性能、高可用、高可靠等质量属性的复杂软件系统 3. 学会分析和构建高效的分布式软件系统  实验要求  1. 2人结对成组 2. 基于实验2-4在不同层面（计算、数据、表示）上的软件架构技术，对遗留进销存系统进行分布式改造。 3. 按照用户需求，商业业务逻辑，将各部分的内容进行逐步整合，形成一个功能完整，且具有较高可用性、可靠性等的分布式软件系统。 4. 分析改造前后系统的非功能指标。 5. 应该出设计过程和实现细节 | | | | |
| 实验内容与过程 | | | | |
| * 1. **对遗留系统从三个层面进行全面的分布式改造的设计过程和实现细节。**   **计算层：**  将项目部署到个人服务器上，并实现计算层分布式改造。应用负载均衡技术改造遗留的“进销存”系统，赋予支持海量用户的在线高并发请求的能力，请给出设计细节并分析负载均衡前后的区别。  首先将进销存系统打包成jar包，然后通过共享文件夹，传入到master结点中，之后使用java -jar app.jar方法，进行测试localhost:8081页面是否可以打开    然后修改master结点的配置文件，slaver结点端口改为8081    在其他两个节点启动jar包  然后访问master 的ip，进行负载均衡    **数据层：**  使用MyCat数据库分库分表技术改造遗留的“进销存”系统。创建具有复杂表结构和含有较大数据量的数据库表。  修改原系统中数据库连接配置：    在server.xml中为系统创建用户：    端口号保持8066：    在schema.xml中为用户配置逻辑数据库（经测试表名不区分大小写）：    配置数据节点：    配置实际数据库连接信息：    读取不能用localhost，一定要127.0.0.1  在rule.xml中配置分片规则：    使用mod-long将数据分为三份以缓解单一数据库压力：    测试：  数据被平均写入三个数据库，数据分库成功              下面尝试另一种分片规则枚举分片  修改表格应用的规则    修改规则使用的参数为地址    为枚举分片配置相关参数，枚举规则文件，参数类型0为数字，非零为字符串，匹配不到时默认第一个节点    在rule.xml文件同一层创建文件    修改文件内容    重启mycat服务测试        添加成功，查看分库，山西录入第二个数据库，江苏录入第三个数据库      **缓存cache：**  添加缓存标签：    启动类添加标签：    添加缓存依赖：    添加缓存空间文件：    **2.2 每次改造的效果的具体体现（运行截图及文字等的论述）是什么？**  **计算层改造性能测试：**  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    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。  **数据层改造性能测试：**  对MyCat分库分表对系统性能的影响进行测试，测试工具同样使用ab。测试分为读写两部分。  先对分片写对系统性能的影响进行测试  在分片的情况下10000条写请求，并发数1000    在分片的情况下100000条写请求，并发数10000    在不分片的情况下10000条写请求，并发数1000    在不分片的情况下100000条写请求，并发数10000    再对从分片数据库读取进行性能测试：  在分片的情况下10000条写请求，并发数1000  看到这里其实属于实验的偶然性，和多次平均测试数据相差较大，选择剔除，并在相同情况下做多次实验以消除实验偶然性    在分片的情况下10000条写请求，并发数1000    在分片的情况下100000条写请求，并发数10000    在不分片的情况下10000条写请求，并发数1000    在不分片的情况下100000条写请求，并发数10000    可以看到共同的趋势是并发数越大单次请求的耗时越长，但是分片与否的差距并不明显，有尝试继续提升并发数，但机器性能受限未能实现。  对缓存的应用对系统性能的影响进行测试  下载ab作为压力测试工具：    修改conf/httpd.conf文件    因为ab装在非启动盘，所以需要添加启动路径以及修改端口号，以及添加存放测试文件夹  下面是添加缓存情况的压力测试结果：    具体数据如下：  D:\Program Files\ab\httpd-2.4.43-lre312-x86-vc14\Apache24\bin>ab -n 2000 -c 500 http://localhost:8081/goods/delete  This is ApacheBench, Version 2.3 <$Revision: 1874286 $>  Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  Licensed to The Apache Software Foundation, http://www.apache.org/  Benchmarking localhost (be patient)  Completed 200 requests  apr\_pollset\_poll: The timeout specified has expired (70007)  Total of 279 requests completed  D:\Program Files\ab\httpd-2.4.43-lre312-x86-vc14\Apache24\bin>ab -n 2000 -c 50 http://localhost:8081/goods/delete  This is ApacheBench, Version 2.3 <$Revision: 1874286 $>  Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  Licensed to The Apache Software Foundation, http://www.apache.org/  Benchmarking localhost (be patient)  Completed 200 requests  Completed 400 requests  Completed 600 requests  Completed 800 requests  Completed 1000 requests  Completed 1200 requests  Completed 1400 requests  Completed 1600 requests  Completed 1800 requests  Completed 2000 requests  Finished 2000 requests  Server Software:  Server Hostname: localhost  Server Port: 8081  Document Path: /goods/delete  Document Length: 41 bytes  Concurrency Level: 50  Time taken for tests: 4.238 seconds  Complete requests: 2000  Failed requests: 0  Total transferred: 292000 bytes  HTML transferred: 82000 bytes  Requests per second: 471.97 [#/sec] (mean)  Time per request: 105.938 [ms] (mean)  Time per request: 2.119 [ms] (mean, across all concurrent requests)  Transfer rate: 67.29 [Kbytes/sec] received  Connection Times (ms)  min mean[+/-sd] median max  Connect: 0 0 0.2 0 1  Processing: 1 105 220.9 22 1259  Waiting: 0 98 220.5 18 1259  Total: 1 105 220.9 22 1260  Percentage of the requests served within a certain time (ms)  50% 22  66% 42  75% 70  80% 98  90% 302  95% 502  98% 1256  99% 1258  100% 1260 (longest request)  关闭缓存进行同样的压力测试    具体数据如下：  D:\Program Files\ab\httpd-2.4.43-lre312-x86-vc14\Apache24\bin>ab -n 2000 -c 50 http://localhost:8081/goods/delete  This is ApacheBench, Version 2.3 <$Revision: 1874286 $>  Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/  Licensed to The Apache Software Foundation, http://www.apache.org/  Benchmarking localhost (be patient)  Completed 200 requests  Completed 400 requests  Completed 600 requests  Completed 800 requests  Completed 1000 requests  Completed 1200 requests  Completed 1400 requests  Completed 1600 requests  Completed 1800 requests  Completed 2000 requests  Finished 2000 requests  Server Software:  Server Hostname: localhost  Server Port: 8081  Document Path: /goods/delete  Document Length: 41 bytes  Concurrency Level: 50  Time taken for tests: 40.504 seconds  Complete requests: 2000  Failed requests: 0  Total transferred: 292000 bytes  HTML transferred: 82000 bytes  Requests per second: 49.38 [#/sec] (mean)  Time per request: 1012.606 [ms] (mean)  Time per request: 20.252 [ms] (mean, across all concurrent requests)  Transfer rate: 7.04 [Kbytes/sec] received  Connection Times (ms)  min mean[+/-sd] median max  Connect: 0 0 0.3 0 2  Processing: 454 972 384.2 878 3558  Waiting: 454 971 384.2 878 3558  Total: 454 972 384.2 878 3558  Percentage of the requests served within a certain time (ms)  50% 878  66% 916  75% 959  80% 999  90% 1140  95% 1441  98% 2760  99% 3229  100% 3558 (longest request)  结果分析：  从最直观的数据，时间来看，开启缓存进行查询时，所需时间为Time taken for tests: 4.238 seconds，而未开启缓存进行压力测试的时候，时间为：Time taken for tests: 40.504 seconds，可以看到，缓存对多线程批量访问的情况下，提升速度效果明显。 | | | | |
| 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 | | --- | --- | --- | --- | --- | | 2020.5.29 | 16:00-18:00 | 任永鹏 | 黄项宇轩 | 熟悉实验内容 | | 2020.5.30 | 9:00-10:00 | 黄项宇轩 | 任永鹏 | 结合之前实验结果 | | 2020.5.30 | 10:00-12:00 | 任永鹏 | 黄项宇轩 | 实现计算层部分 | | 2020.5.30 | 15:00-22:00 | 黄项宇轩 | 任永鹏 | 实现数据层 | | 2020.5.31 | 8:00-12:00 | 任永鹏 | 黄项宇轩 | 实现表示层 | | 2020.5.31 | 15:00-18:00 | 黄项宇轩 | 任永鹏 | 整理报告 |   【注意】该表格可自行增加更多的行。 | | | | |
| **（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 | | --- | --- | --- | --- | | 2020.5.30 | 分页修改出现问题 | 修改后端对应接口 |  | | 2020.5.30 | Ab安装 | 相互交流 |  | | 2020.5.31 | apache 无法启动 no listening sockets available 错误 | 去该apache的端口，在httpd.conf里面该，可以改成其他端口 |  | | 2020.5.31 | 异常:This application has no explicit mapping for /error, so you are seeing this as a fallback. | 控制器的url访问路径与注解@GetMapping（"/xxxx"）不匹配 |  | | 2020.5.31 | MySQL出现MySQLNonTransientConnectionException: No operations allowed after statement closed异常的解决办法 | Mysql链接后面添加?autoReconnect=true |  | | 2020.6.01 | 压力测试实验存在偶然性 | 多次实验消除偶然性 |  | | 2020.6.01 | 压力测试时并发数10000+ | 机器性能受限 |  | | | | | |
| **（3）结对开发工作现场照片** | | | | |
|  | | | | |
|  | | | | |
| 实验总结 | | | |  |
| 本次实验涵盖了本学期几乎所有的实验，这次实验让我们对本学期的所有实验有了一个清晰的定位和重新的认识。充分运用本学期所学习使用的所有技术，可以让我们的代码能力得到进一步的提高。一个学期完整的实验流程下来，上学期遗留的进销存系统在新技术的应用下脱胎换骨，让一个本来是为了符合实验要求而构建的系统，一步步地能够在实际环境下使用，每一项新技术的加入，都让系统更具实用性和活力，最后的整合甚至在系统各个层面的性能上做了改造。相信这些实验过程中掌握的新知识和新技术一定是未来学习工作中起到基石一般重要的作用。分库分表操作可以让一个项目不局限于一个数据库，缓存cache可以压缩项目在高并发场景下的反应时间。  最后感谢本课程所有老师！ | | | | |
| 教师评语 | | | |  |
|  | | | | |