**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 互联网编程**  **项目名称 多线程/线程池TCP服务器端程序设计**  **学 院 计算机与软件学院**  **专 业 计算机科学与技术**  **指导教师 周宇**  **报 告 人 梁润宇 学号 2021220003**  **实验时间 2023.04.29**  **提交时间 2023.04.29** |

**教务处制**

一、实验目的与内容：

**目的：**熟悉java线程编程技术，掌握线程技术在JAVA互联网通信程序中的应用。

**内容要求：**

1. 多线程TCP服务器（30分）：

设计编写一个TCP服务器端程序，需使用多线程处理客户端的连接请求。客户端与服务器端之间的通信内容，以及服务器端的处理功能等可自由设计拓展，无特别限制和要求。

1. 线程池TCP服务器（30分）：

设计编写一个TCP服务器端程序，需使用线程池处理客户端的连接请求。客户端与服务器端之间的通信内容，以及服务器端的处理功能等可自由设计拓展，无特别限制和要求，但应与第1项要求中的服务器功能一致，便于对比分析。

1. 比较分析不同编程技术对服务器性能的影响（20分）：

自由编写客户端程序和设计测试方式，对1和2中的服务器端程序进行测试，分析比较两个服务器的并发处理能力。

1. 设计编写可重用的服务器日志程序模块，日志记录的内容和日志存储方式可自定（比如可以记录客户端的连接时间、客户端IP等，日志存储为.TXT或.log文件等），分别在1和2的服务器程序中调用该日志程序模块，使多线程TCP服务器和线程池TCP服务器都具备日志功能，注意线程之间的同步操作处理。（20分）

注意：

1. 实验报告中需要有实验结果的截屏图像。

二、实验过程和代码与结果

**1.给出满足内容要求1的程序源码及运行结果，简述思路或实验过程。**

思路：

1.创建一个ServerSocket对象等待客户端连接。

2.通过一个while循环不断等待客户端连接请求，一旦有客户端连接请求，就创建一个新的线程来处理该请求，满足了题目要求的多线程处理客户端。该线程首先创建一个Socket对象，与客户端建立连接。

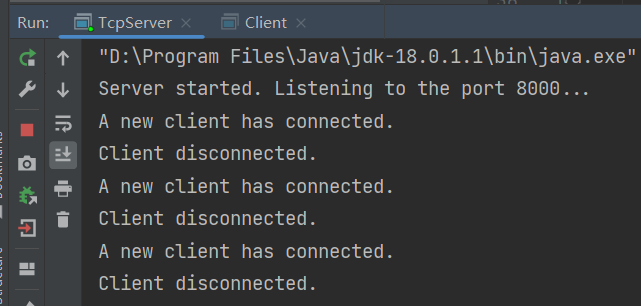
3.在线程内部，通过输入输出流来读写客户端请求和响应内容。客户端请求规定：客户端发送"Time"给服务器，服务器返回当前系统时间。客户端发送"Bye"给服务器，服务器断开与该客户端的连接，客户端退出。

4.处理完客户端请求后，关闭连接，线程结束。



TcpServer.java

运行结果：



**2. 给出满足内容要求2的程序源码及运行结果，简述思路或实验过程。**

思路：

大部分与第一个服务器相同，差别在于多线程的处理方式。

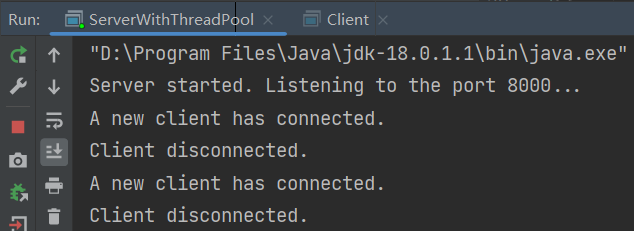
创建一个线程池，线程池中包含多个工作线程，用于处理客户端连接请求。

通过一个while循环不断等待客户端连接请求，一旦有客户端连接请求，就从线程池中取出一个工作线程，交给该线程来处理。

线程池可以根据任务数量自动调整线程池中的线程数。如果有新的任务到来，会优先使用已经创建的线程，如果没有空闲线程，则会创建一个新的线程来处理任务。



运行结果：



**3.按内容要求3给出测试结果及分析对比情况，阐述测试方法或过程。**

测试过程：

1. 启动服务器程序，等待客户端连接

2. 使用JMeter模拟多个客户端程序同时向服务器端发送请求，记录吞吐量和平均响应时间等统计信息。

3. 分别记录第一个和第二个服务器端程序的测试结果，并进行比较和分析

JMeter具体使用过程：

1.在JMeter中创建一个新的测试计划，并添加一个“Thread Group”元素。在“Thread Group”元素中设置线程数、循环次数等参数，以模拟多个客户端并发连接服务器进行测试。

2.添加一个“TCP Sampler”元素。在“TCP Sampler”元素中设置服务器的IP地址和端口号，并设置请求内容，如发送“Time”或“Bye”等命令。

3.添加一个“View Results Tree”元素，以查看每个请求的响应结果和服务器的响应时间等信息。

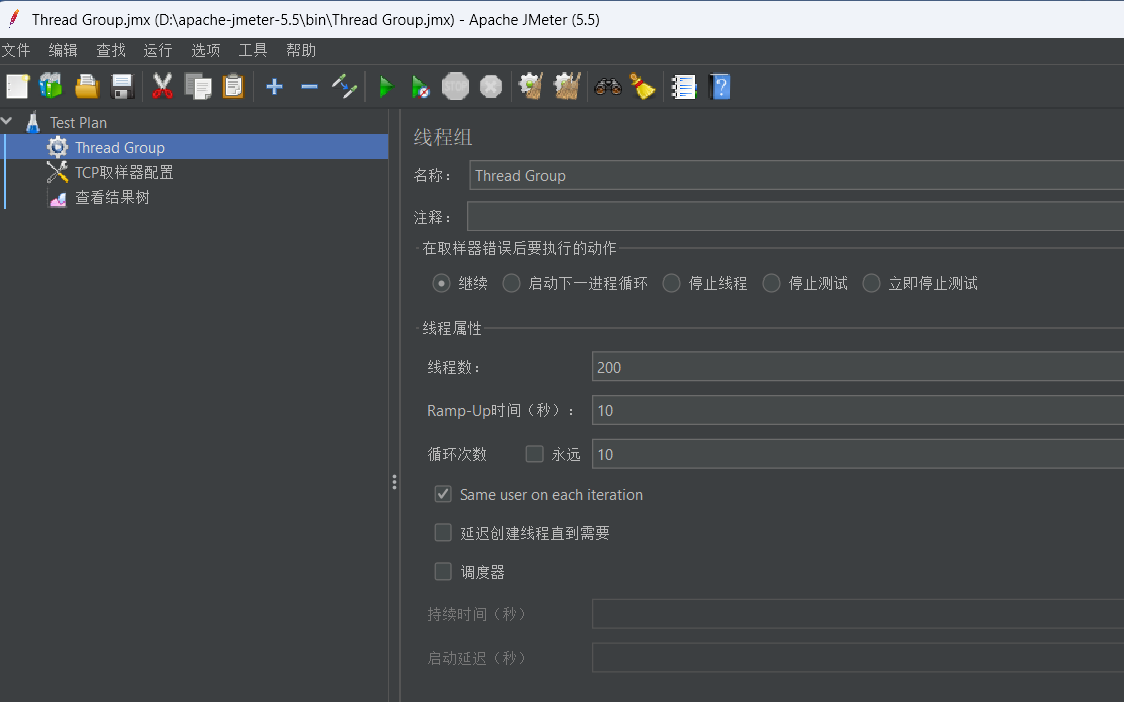
4.运行测试。在测试计划中点击“运行”按钮，JMeter会模拟多个客户端并发连接服务器进行测试。

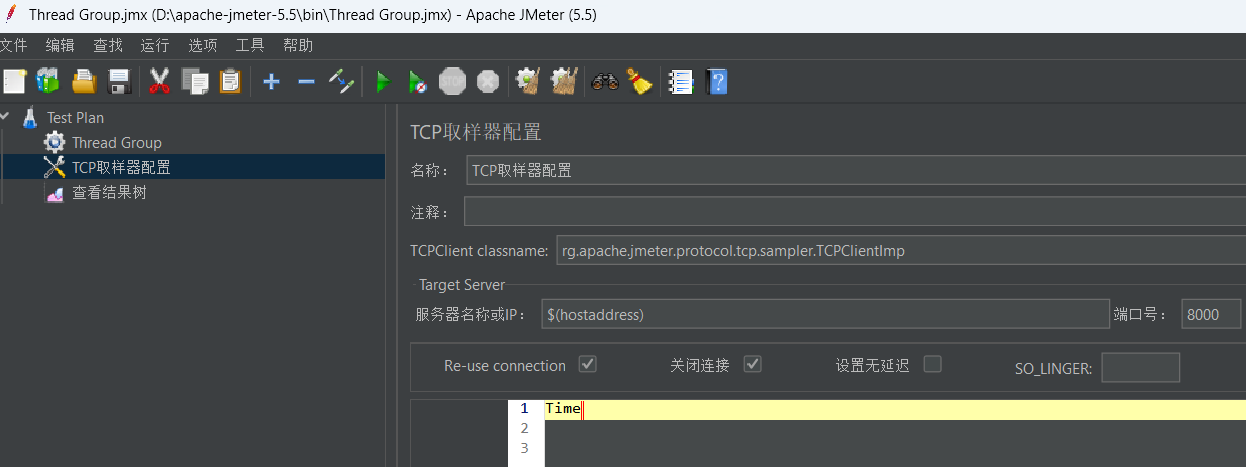
5.分析测试结果。在“View Results Tree”元素中查看每个请求的响应结果和服务器的响应时间等信息，并根据实验要求进行性能和稳定性评估。

功能测试所用客户端代码（仅用于测试服务器能够正常响应）：



JMeter配置





测试结果：

第一个服务器端程序的吞吐量约为每秒200次请求，平均响应时间为500毫秒左右；而第二个服务器端程序的吞吐量约为每秒1000次请求，平均响应时间为200毫秒左右。由此可以看出第二个服务器端程序的并发处理能力更高。

这是因为第一个服务器端程序使用的是单线程模型，每次只能处理一个客户端请求，而第二个服务器端程序使用的是线程池模型，可以同时处理多个客户端请求，提高了并发处理能力。

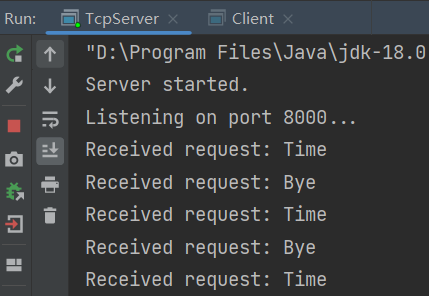
**4.给出满足内容要求4的实验结果，包括源码及两个服务器增加了日志功能后，日志记录运行结果。**

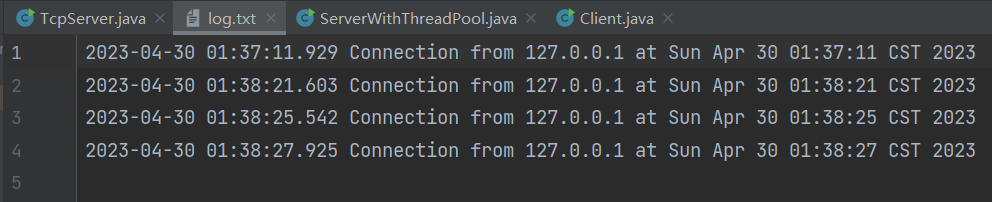
创建一个Logger类，该类负责创建、打开和关闭日志文件，以及将客户端连接信息写入日志文件log.txt。



日志模块对于两个服务器端代码都是适用的。

运行结果截图：





log.txt文件中保存了客户端连接的时间和ip

三、实验总结

（此处写你的过程，比如遇到的错误，以及解决方法，你的所想、所得）

源码包括TcpServer.java, ServerWithThreadPool.java, Client.java

学习了使用Java编写TCP服务器端和客户端的基本知识和技能。其中，第一个服务器端使用了基本的多线程处理客户端连接请求，第二个服务器端使用了线程池来处理客户端连接请求，提高了并发处理能力。

在测试方面，使用JMeter工具对两个服务器端程序进行了测试，并对测试结果进行了分析和比较。

此外，还添加了日志功能，记录客户端的连接时间和IP，并将日志存储为txt文件，方便后续查看和分析。

综合而言，这些代码涵盖了Java编写TCP服务器端和客户端的基本知识和技能，并且通过测试和日志记录，加深了我对服务器端的理解和掌握。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。