**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 互联网编程**  **项目名称 传输协议与套接字应用编程**  **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 2023.06.20**  **提交时间 2023.06.20** |

**教务处制**

一、实验目的与内容：

**目的：**

1. 掌握服务器端Socket编程开发能力。

2.掌握HTTP协议并能用于实际编程开发。

3.掌握cookie编程技术，会应用于在web应用开发中持久存储和传递客户端状态，会编程管理cookie。

**内容要求：**

基于java编程实现一个HTTP服务器程序（20分）和HTTP客户端程序（15分），要求

1. 采用多线程技术或线程池编程技术处理客户端请求，支持多客户端同时访问；（10分）
2. 实现GET、HEAD和POST请求，对客户端发送的不同请求给予正确响应；（15分）
3. 在服务器上放一个静态网站（由HTML文本、图片文件或JS文件等组成），能根据不同请求，返回包括文本和图像2种（及以上）类型的响应，客户端可以正确显示和访问。（10分）
4. HTTP客户端程序能与该HTTP服务器连接并展示响应结果，正确发送不同类型的请求。（10分）
5. 能使用cookie编程技术保存和传递会话状态信息，比如保存用户信息等，需要保存的信息可自行决定。（10分）
6. 对服务器进行性能分析。对服务器进行压力测试，测试可支持多少个客户端同时访问，测试可支持多少个文件同时传输等。（10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.给出满足内容要求的HTTP服务器端程序源码和HTTP客户端程序源码，简述设计思路或实验过程。**

**设计思路：**

**1.1 线程池处理客户端请求，支持多客户端同时访问**

首先，创建了一个大小为10的线程池对象，线程池大小在运行过程中可以动态调整。



在服务器的主循环中，通过serverSocket.accept()方法等待客户端的连接请求。一旦有客户端连接请求到达，就会创建一个新的Socket对象来处理该连接。

徽标

低可信度描述已自动生成

然后，将该Socket对象传递给线程池的submit()方法，以便在线程池中执行处理逻辑



在RequestHandler线程类中，实现了Runnable接口，并重写了run()方法。该方法就是处理客户端请求的逻辑。



**1.2. 实现GET、HEAD和POST请求**

①GET请求：

服务器首先解析请求行中的方法、路径等信息，并获取到请求的路径。

根据请求路径，判断是否为静态网站的文件。如果是文件，则读取文件内容并构造响应。

构造HTTP响应，包括响应头和响应内容，将响应发送给客户端。

文本

中度可信度描述已自动生成

②HEAD请求：

服务器解析请求行中的方法、路径等信息，并获取到请求的路径。

根据请求路径，判断是否为静态网站的文件。如果是文件，则构造响应头，不包含响应内容。

构造HTTP响应，只包括响应头部分，将响应发送给客户端。

文本

描述已自动生成

③POST请求：

服务器解析请求行中的方法、路径等信息，并获取到请求的路径。

根据请求路径，执行相应的业务逻辑，比如处理表单提交等操作。

构造HTTP响应，包括响应头和响应内容，将响应发送给客户端。

文本

中度可信度描述已自动生成

**1.3. 在服务器上放一个静态网站**

确定服务器上放置静态网站的目录结构，包含HTML文本、图片文件等资源。

当收到客户端的请求时，解析请求的路径，获取要访问的资源文件路径。

根据资源文件的扩展名判断其类型，例如.html为文本类型，.jpg为图像类型等。

根据资源文件的类型构造相应的响应头，包括Content-Type字段，指定返回的数据类型。

读取资源文件的内容，并构造响应实体。

将响应头和响应实体发送给客户端。

图片包含 文本

描述已自动生成

**1.4使用cookie保存和传递会话状态信息**

在服务器端，当收到客户端的登录请求时，可以生成一个唯一的会话标识符（例如使用UUID），并将其存储在服务器端的会话管理器中，同时将该标识符作为一个Cookie发送给客户端。

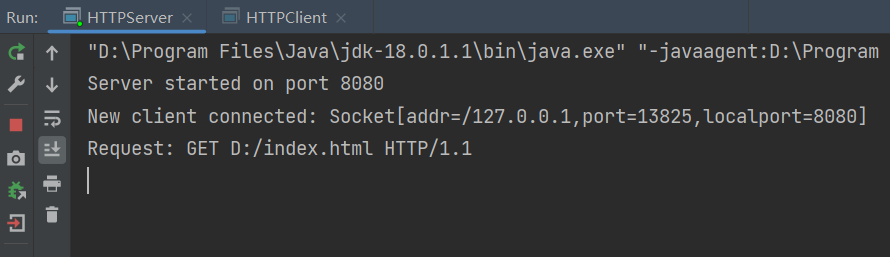


在客户端，收到服务器返回的包含会话标识符的Cookie后，将会话标识符存储在客户端，以备将来的请求中使用。



**2. 给出满足不同内容要求所进行测试的运行结果，包括客户端的运行结果及服务器端的运行结果，并简述所进行的测试。**

客户端向服务器端发送GET, HEAD, POST请求



服务器端

手机屏幕截图

描述已自动生成

文本

描述已自动生成

客户端

**3.服务器性能分析总结。**

**对服务器进行压力测试，测试可支持多少个客户端同时访问，测试可支持多少个文件同时传输等，描述测试方法与测试结果。**

使用JMeter进行压力测试。

1. 启动服务器程序，等待客户端连接

2. 创建测试计划：打开 JMeter，创建一个新的测试计划。在测试计划中，定义测试的目标、配置线程组、添加取样器（Sampler）和监听器（Listener）等。

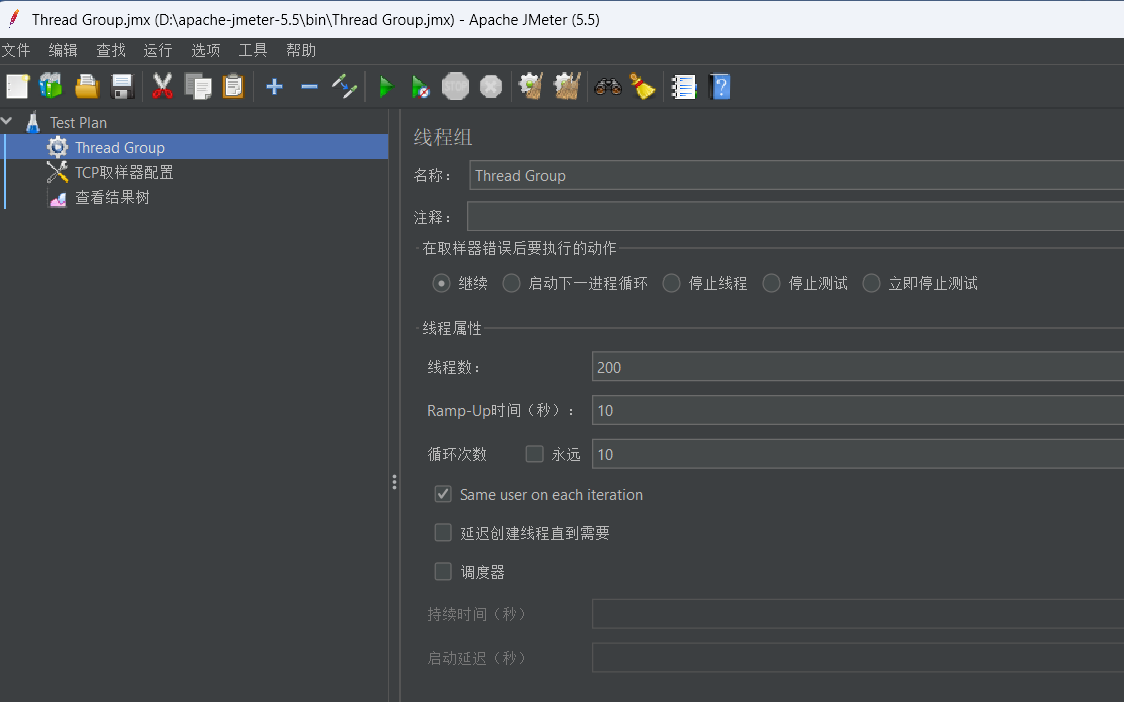
3.配置线程组：线程组定义了并发用户的行为。在线程组中，设置用户数目、循环次数、启动延迟等参数。您可以根据需要添加多个线程组以模拟不同的用户行为。

4.添加 HTTP 请求：在线程组下，添加一个 HTTP 请求取样器。配置该取样器以发送适当的请求，例如 GET、POST 或 HEAD 请求。指定服务器的主机和端口，并设置请求路径、请求头、请求体等。

5.添加监听器：为了收集和分析测试结果，我们添加一个或多个监听器。监听器可以显示响应时间、吞吐量、错误率等指标，并提供图表和报告。一些常用的监听器包括查看结果树（View Results Tree）、聚合报告（Aggregate Report）等。

6.配置测试参数：根据需要，可以配置其他测试参数，如并发用户数、请求频率、持续时间等。

7.运行测试：保存测试计划，并点击 JMeter 工具栏中的“运行”按钮以启动压力测试。JMeter 将模拟并发用户发送请求到服务器，并记录响应时间、吞吐量和错误信息等。

8.分析结果：在测试运行完成后，查看监听器中的结果数据，生成图表或报告，以评估服务器的性能和稳定性。

图形用户界面, 文本

描述已自动生成

测试结果：服务器可以承受200个客户端同时连接，可以承受30个200kb大小的文件同时传输。

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）

在编写HTTP客户端和服务器端程序的过程中，我们涉及了多线程编程、HTTP协议的处理、静态文件的响应、会话状态的管理等关键概念和技术。下面是对整个过程的总结思考：

1.多线程技术：使用多线程或线程池编程技术可以有效地处理多个客户端的并发请求。通过为每个客户端请求创建一个独立的线程或从线程池中获取可用线程，可以实现多客户端同时访问的支持。

2.HTTP协议处理：根据HTTP协议的规范，我们编写了客户端和服务器端的代码来处理GET、HEAD和POST请求。在客户端，我们使用Socket与服务器建立连接，并通过发送HTTP请求头和接收HTTP响应来进行通信。在服务器端，我们根据请求类型和请求路径，返回相应的静态文件或执行相应的操作。

3.静态文件响应：服务器端程序可以根据不同的请求路径，返回对应的静态文件作为响应。通过读取文件内容，并将其作为响应的一部分发送给客户端，我们可以实现在客户端正确显示和访问包括文本和图像等多种类型的响应。

4.会话状态管理：为了保存和传递会话状态信息，我们可以使用cookie编程技术。在服务器端，可以生成唯一的session ID，并将其存储在cookie中发送给客户端。客户端接收到cookie后，将其保存，并在后续请求中将cookie带回服务器。服务器可以根据session ID来识别用户，并恢复其相关的会话状态信息。

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：周宇  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。