文章编号:1005－0523(2007)05－0110－03

用 Java 实现流式 Socket 通信

刘邦桂,李正凡

(华东交通大学 信息工程学院,江西 南昌 330013)

摘要:基于 Java 的多线程机制和流式 Socket 的网络编程是设计并行网络通信程序的一种有效方法,文中介绍了 Socket 的通信机制,详细分析了 Java 语言中的 Socket 编程的步骤和方法,最后用 Java 实现了流式 Socket 通信.

关 键 词:Java;Socket;多线程;通信机制
中图分类号:TP311　　**文献标识码**:A

1 引言

在 Internet 被广泛使用的今天,网络编程就显得更加重要了,网络应用是 Java 语言取得成功的领域之一,Java 已经成为 Internet 上最为流行的一种编程语言.Java 语言的网络功能非常强大,其网络类库不仅使我们可以开发访问 Internet 应用层程序,而且还可以实现网络底层的通信.在 TCP/IP 网络中,不同的机器之间通信时,数据的传输是由传输层控制的,这包括数据要发往的目标机器及应用程序数据的质量控制等.TCP/IP 网络中最常用的传输协议 TCP 和 UDP,传输层通常以 TCP 和 UDP 协议来控制端点到端点的通信.用于通信的端点是由 Socket 来定义的,而 Socket 是由 IP 地址和端口号组成.常用的 Socket 类型有两种:流式 Socket 和数据报式 Socket.流式是一种面向连接的 Socket,针对于面向连接的 TCP 服务应用;数据报式 Socket 是一种无连接的 Socket,对应于无连接的 UDP 服务应用.本文详细介绍了流式 Socket 通信的基本原理,然后在这些原理的基础上,用 Java 实现了一个多用户网上聊天系统.

2 Java 中流式 Socket 通信类

Socket 类和 ServerSocket 类是用 Java 实现流式 Socket 通信的主要工具,在编程实现的过程中创建一个 ServerSocket 对象就创建了一个监听服务,创建一个 Socket 对象就建立了一个 Client 与 Server 间的连接.

2.1 ServerSocket 类

下面的语句将创建一个 ServerSocket 对象,同时在运行该语句的计算机的指定端口处建立一个监听服务:

```
ServerSocket MyListener＝new ServerSocket(8000);
```

这里指定提供监听服务的端口号是 8 000,一台计算机可以同时提供多个服务,这些不同的服务之间通过端口号来区别,不同的端口号可以提供不同的服务,Client 连接到哪个端口,就可以接受哪个端口提供的服务.为了随时监听可能的 Client 请求,还应该执行如下的语句:

```
Socket LinkSocket ＝ MyListener.accept();
```

这个语句调用了 ServerSocket 对象的 accept()方法,这个方法的执行将使 Server 端的程序处于等待状态,程序将一直阻塞直到捕捉到一个来自 Client 端的请求,并返回一个用于与该 Client 通信的 Socket 对象 LinkSocket.此后 Server 程序只要向这个 Socket 对象读写数据,就可以实现向远端的 Client 读写数据.

需要结束监听时,只需用如下的语句关闭 ServerSocket 对象:MyListener.close();

2.2 Socket 类

当 Client 程序需要从 Server 端获取信息及其他服务时,应该创建一个 Socket 对象:

```
Socket MySocket＝new Socket ("LBG",8 000);
```

Socket 类的构造函数有两个参数,第一个参数是欲连接到的 Server 计算机的主机地址,第二个参数是该 Server 机上提供服务的端口号.Socket 对象建

收稿日期:2007－09－14

作者简介:刘邦桂(1983－),男,江西赣州人,华东交通大学信息工程学院在读硕士研究生,研究方向:数据库.

立成功之后，就可以在 Client 和 Server 之间建立一个连接，并通过这个连接在两个端点之间传递数据.

3　流式 Socket 的通信机制

流式 Socket 所完成的通信是一种基于连接的通信，即在通信开始之前先由通信双方确认身份并建立一条专用的虚拟连接通道，然后它们通过这条通道传送数据信息进行通信，当通信结束时再将原先所建立的连接拆除. 这个过程如图 1 所示，其中 Server 端首先在某端口提供一个监听 Client 请求的监听服务并处于监听状态，当 Client 端向该 Server 的这个端口提出服务请求时，Server 端和 Client 端就建立了一个连接和一条传输数据的通道，当通信结束时，这个连接通道将被同时拆除.

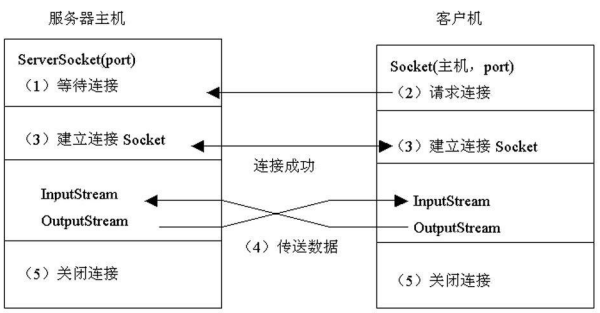

图 1　流式 Socket 通信过程

4　Socket 通信的应用

结合上面的 Socket 通信的基本原理，下面用 java 实现了流式 Socket 通信多用户聊天系统. 从总体来看，这是一个支持多 Client 的 Socket 通信的聊天系统，根据设计过程中服务器和客户连接的方式，有两种解决的方案：

● 方案一：在一台计算机上启动多个服务器程序，而指定服务器监听端口号不同，一个服务器对应一个客户机.

●方案二：将服务器写成多线程的，不同的处理线程为不同的客户服务. 主线程只负责循环等待，处理线程负责网络连接，接收客户输入的信息.

考虑到第一种方案，如果要启动多个客户的话，要占用的监听端口号多，通信比第二种方案更不方便，所以下面我是采用第二种方案来实现的.

4.1　聊天服务器端

聊天服务端的主要任务有两个：一是监听某端口，建立与客户的 Socket 连接，处理一个客户的连接后，能很快再进入监听状态；二是处理与客户的通信，由于聊天是客户之间进行，所以服务器的职责是将客户发送的消息转发给其他客户. 为了实现两个目标，必须设法将任务分开，可以借助多线程技术，在服务方为每个客户连接建立一个通信线程，通信线程负责接受客户的消息并将消息转发给其他客户. 这样主程序的任务就简单化，循环监听客户连接，每个客户连接成功后，创建一个通信线程，并将与 Socket 对应的输入输出流传给该线程. 程序代码如下：

```
public class talkserver
{ public static Client[] allclient=new Client[20];//存放所有通信线程
  public static int clientnum=0;//连接客户数
  public static void main(String args[])
{ try {ServerSocket s=new ServerSocket(8000);//建立监听服务
  while (true)
    { Socket s1=s.accept();//等待客户连接
      DataOutputStream dos = new DataOutputStream(s1.getOutputStream());
      DataInputStream din = new DataInputStream(s1.getInputStream());
      allclient[clientnum]= new Client(clientnum,dos,din);
      allclient[clientnum].start();//创建与客户对应的通信线
      clientnum++;
      }}catch(IOException e){ }
  }}
class Client extends Thread
    { int id;//客户标识
  DataOutputStream dos;//去往客户的输出流
  DataInputStream din;//来自客户的输入流
      public Client (int id, DataOutputStream dos, DataInputStream din)
  {this.id=id;this.dos=dos;this.din=din;}
  public void run()
  { while(true)//循环读取客户数据转发给其他客户
    {try{ int m=talkserver.clientnum;//客户数
      String message="第"+(id+1)+"个客户发言:"+din.readUTF();//读客户数据，无数据时线程挂起
      for (int i=0;i<m;i++){
          talkserver.allclient[i].dos.writeUTF(message);//转发给其他客户
        }} catch(IOException e){}
    }}}
```

每个通信线程均在循环检测是否本线程对应的客户有数据发送过来，一旦接收到数据就通过循环将数据发送到所有客户(包括自己)的 Socket 通道.

4.2 聊天客户端

客户端的职责也有两个：一是能提供一个图形界面实现聊天信息的输入和显示，其中包括处理用户输入事件；二是要随时接收来自其他客户的信息并显示出来.因此，在客户端也采用多线程实现，应用程序主线程负责图形界面的输入处理，而接收消息线程负责读取其它客户发来的数据.

每个用户首先向服务器请求连接后，才能与其他客户进行连接.所以对应服务器端端口等待连接，在客户端要请求服务器连接，也即先实例化 Socket 类，传入主机的 ip 地址和服务器端口号.

```
Socket s1 = new Socket("LBG",8 000);
```

主机接收到请求后，建立连接，服务端产生一个 Socket 实例，同该客户端建立通信，服务器端和客户端同时打开输入和输出流进行数据的传输.

```
DataInputStream dis=new DataInputStream(s1.get-
InputStream());
final DataOutputStream dos=new DataOutputStream
(s1.getOutputStream());
```

当客户要发送数据的时候，先将数据发送到服务器，然后由服务器转发给其他正在检测的用户，下面代码就是主要负责向其他客户发送数据.

```
public void actionPerformed(ActionEvent e) {
  try{
    dos.writeUTF(""+input.getText());//发送数据
  } catch(IOException z) {}} } );
```

客户端必须时刻检测服务器端有无数据数据发出，如果检测到其他客户发送了消息后，接收消息线程读取网络消息，显示在聊天界面中.

```
try {
        String str = new String(dis.readUTF());//
读取其他客户经服务器转发的消息
    displayarea.append(str+"/n");//将消息添加到
文本域显示
        } catch (IOException e ){ }
```

4.3 聊天系统的运行界面

图 2 是聊天系统的运行界面，其中第一个是服务器，其他是三个不同的客户.

图 2 系统运行界面

5 结论

这是一个流式 Socket 通信的应用，实现了网上多用户聊天，在系统中还有许多方面值得改进，例如：如何修改服务方，使用户自己发送的消息不显示在自己的文本域中；增加一个用户名输入界面，用户输入身份后再进入聊天界面；在客户方显示用户列表，可以选择将信息发送给哪些用户.这些方面得到改进后就可以极大地提高其应用价值.

参考文献：

[1] Scottt Oaka，Henry Wong 著 .Java 线程[M]. 南京：东南大学出版社，2006.
[2] 赵毅 .Internet 网络低层编程及 Java 实现[J]. 上海工程技术大学学报，2004，18(4)：332－335.
[3] 胡多勋，文富荣，汪自云 .Java 网络通信机制的研究[J]. 湖北师范学院学报，2003，21(3)：40－44.
[4] Cay S.Horstmann，Gary Cornell 等著 .Java 核心技术[M]. 北京：机械工业出版社，2006.
[5] Chad Darby，John Griffin，Pascal han 等著.邱仲潘，等译.Java 网络编程[M]. 北京：电子工业出版社，2002.

Implementation of Socket Stream Communication in Java

LIU Bang-gui，LI Zheng-fan

(School of Information Engineering，East China Jiaotong University，Nanchang 330013，China)

Abstract: The network programming based on multithreading mechnism and socket stream in Java is a efficient method.The article mainly introduces the socket communication mechanism，makes a detailed analysis on the step and method of socket programming in Java language and also gives an application example.

Key words: Java；socket；multithreading；communication mechanism