

文章编号: 1005-8451 (2008) 01-0020-03

铁路局车站间数据交换系统的设计与实现

王 斌

(北京交通大学 计算机与信息技术学院, 北京 100044)

摘 要: 在铁路运输生产中, 车站起着重要的作用。讨论铁路局调度系统中铁路局与车站数据交换的实现方法, 利用 Socket 技术使得数据交换变得准确和快捷。实践表明, 该系统能够很好地满足日常运输生产要求。

关键词: 调度系统; 车站; 数据交换系统; 设计

中图分类号: U294.1-39 **文献标识码:** A

Design and implementation of Data Switch System between stations of railway administration

WANG Bin

(Institute of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: It was well-know that stations played a very important role in entire railway transportation activities. In this paper, the data exchange implementation methods between railway administration and stations were discussed in BDS(Bureau Dispatch System). The data exchange was facilitated by using the Socket. Furthermore, some pivotal Java pseudocode was also given. It was proved that the System was satisfied the requirements of daily transportation works.

Key words: Dispatch System; station; Data Switch System; design

在铁路三级调度指挥体系中, 铁路局总调室作为关键的一级指挥机构, 担负着组织日常运输生产和确保运输安全的重要任务, 车站扮演着运输生产主要完成者的角色。在一个生产环节中, 铁路局调度人员通过编制各种运输生产计划指挥站段开展工作, 并根据站段上报的实时信息及时调整计划, 以保障各项运输任务的顺利完成, 在铁路局调度与车站作业人员间存在着大量的信息需要进行广泛、及

时和准确的交换。

在传统的作业模式中, 铁路局调度、车站作业人员间的信息交换主要利用电话完成信息上传下达, 一方面加重了作业人员的工作负担, 另一方面也容易造成信息交换不及时和不准确等问题。在铁路局车流计划系统的全路推广过程中, 为减轻作业人员劳动强度, 解决信息交换中存在的问题, 设计了基于技术车站车流报告系统, 主要提供车站上报站存车、接收日(班)计划和上报行车时刻点等功能。

收稿日期: 2007-06-04

作者简介: 王 斌, 在读硕士研究生。

(2) 消除不必要的全表扫描。正确使用索引, 避免使用“*”查询;

(3) SQL关键字的正确使用。如果有可能的话, 有些关键字要避免使用, 如: 用 EXISTS 替代 IN、用 NOT EXISTS 替代 NOT IN; 用 EXISTS 替换 DISTINCT; 用 UNION、IN 替换 OR; 用 UNION-ALL 替换 UNION; 用 Where 替代 ORDER BY;

(4) 避免使用耗费资源的操作。带有 DISTINCT、UNION、MINUS、INTERSECT、ORDER BY 的 SQL 语句会启动 SQL 引擎, 执行耗费资源的排序(SORT)功能。

4 结束语

经过以上设计的铁道部货调数据库, 再由 DBA 根据应用情况进行 Oracle 系统的调优, 系统一直以来都在平稳高效地运行。

参考文献:

- [1] 飞思科技产品研发中心. Oracle9i 数据库高级管理[M]. 北京: 电子工业出版社, 2002.
- [2] [美] Dan Hotka. Oracle8i Web 开发指南[M]. 唐文, 李家菁, 王松. 北京: 清华大学出版社, 2002.

1 Socket 基本原理

客户机与服务器之间使用的大部分通讯组件都是基于 Socket 接口来实现的。Socket 是两个程序之间进行双向数据传输的网络通讯端点，由一个地址和一个端口号来标识。每个服务程序在提供服务时都要在一个端口进行，使用该服务的客户机也必须连接该端口。

1.1 服务器端的实现

(1) 首先调用 ServerSocket 类，以某个端口号为参数，创建一个 ServerSocket 对象，即是服务器端的服务程序在该指定端口监听的 Socket；

(2) 服务器端程序使用 ServerSocket 对象的 accept() 方法，接收来自客户机程序的连接请求，此时服务器端将一直保持停滞状态，直到收到客户端发来的连接请求，该方法将返回一个新建的 Socket 类的实例，代表与客户机建立的通讯链路在服务程序内的通讯端点。如果采用 Java 的多线程编程方法，可以实现并发服务器，继续监听来自其他客户的连接请求；

(3) 使用新建的 Socket 对象创建输入和输出流 (InputStream、OutputStream) 对象；

(4) 使用流对象的方法完成服务器与客户端之间的数据传输，按约定协议识别并处理来自客户端的请求数据，并把处理的结果返回给客户端；

(5) 客户端工作完毕后，则服务器端程序关闭与客户端通讯的流和通讯的 Socket；

(6) 在服务器程序运行结束之前，应当关闭用来监听的 Socket。

1.2 客户端的实现

(1) 首先调用 Socket 类的构造函数，以服务器指定的 IP 地址或指定的主机名和指定的端口号为参数，创建一个 Socket 流，在创建 Socket 流的过程中包含了向服务器请求建立通讯连接的过程实现；

(2) 建立了客户端通讯 Socket 后，就可以使用 Socket 的方法 getInputStream() 和 getOutputStream() 来创建输入 / 输出流。这样，使用 Socket 类后，网络输入输出也转化为使用流对象的过程；

(3) 使用输入输出流对象的相应方法读写字节流数据，因为流连接着通讯所用的 Socket，Socket 又是与服务器端建立连接的一个端点，因此，数据将通过连接从服务器得到或发向服务器。这时可以对字节流数据按客户端和服务器之间的协议进行处

理，完成双方的通讯任务；

(4) 待通讯任务完毕后，用流对象的 close() 方法来关闭用于网络通讯的输入输出流，再用 Socket 对象的 close() 方法来关闭 Socket。

2 调度台与车站通信的实现方法

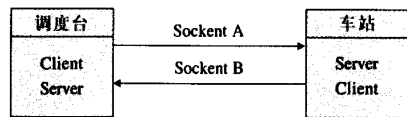


图 1 调度台与车站信息传播示意图

2.1 调度台

(1) 建立启动 Socket 服务。启动应用程序时，指定服务端口 (port)，建立启动 Socket 服务，等待客户端请求。应用程序关闭时，终止其 Socket 服务；

(2) 建立启动 Socket。调度台向车站发送数据信息时，首先以车站为 Server 端，以调度台为 Client 端，建立启动 Socket 连接进行通信；一旦 Socket 连接建立启动，除非 Socket 连接出现异常，不能将其中断。如果此 Socket 连接出现异常而中断，不需要重新建立启动 Socket 连接，只有在调度台需要向车站发送信息时，才重新建立启动 Socket 连接。为了区别车站为 Client 建立的 Socket，此方式建立的 Socket 定义为 SocketA；

(3) 发送数据信息。调度台通过 SocketA 向车站发送数据信息包。调度台必须保证数据信息包成功发送，在发送过程中，当 SocketA 连接出现异常而中断或关闭时，必须重新向车站发送信息；

(4) 接收数据信息。调度台通过 SocketB 接收数据信息包。调度台根据数据包的类型进行相应的处理，然后向车站返回接收到的信息 (包括处理结果)。

2.2 车站

(1) 建立启动 Socket 服务。在启动应用程序时，同时建立启动 Socket 服务，指定服务端口 (port)，只有应用程序关闭时，才终止其 Socket 服务；

(2) 建立启动 Socket。车站向调度台发送数据信息时，首先以调度台为 Server 端，以车站为 Client 端，建立启动 Socket 连接进行通信；一旦 Socket 连接建立启动，除 Socket 连接出现异常，不能进行关闭；

(3) 发送数据信息。车站通过 SocketB 向调度台发送数据包。车站必须保证数据信息包成功发

送。在发送过程中，当 SocketB 连接出现异常而中断或关闭时，必须重新向调度台发送信息；

(4) 接收数据信息。车站通过 SocketA 接收数据信息包。车站根据数据包的类型进行处理，然后向调度台返回接收信息（包括处理结果）。

3 具体实现

在实现的过程中，采用 Java 为开发语言，为了适应全路各局不同的网络条件，采用双向双通道来保证铁路局调度系统与车站报告系统联接的稳定性。铁路局调度系统中，铁路局与车站的计算机互为服务端，又互为客户端。

(1) 服务端 SocketServer 的设计与实现

```
Public class TranServer implements Runnable,
ReadCallback,java.awt.event.ActionListener {
    private Thread thread;
    public static int DEFAULT_PORT = 6868;
    public java.net.ServerSocket listen_socket;
    public java.util.Vector connections;
    private ReadCallback listener;
    private IndicatorLight lights[];
    public boolean isRun = true;
    private javax.swing.Timer timer;
    public synchronized void dataReady(Object obj) {}
    public void run() {}
    public synchronized void socketException(Object
source) {}
    public void start_server() {}
    private synchronized void initConnection(java.net.
Socket client_socket) {}
```

服务端
接受监听线程
服务端口占用的端口号
socket服务端
客户端的连接池
接受信息调用
连接信号灯
服务端运行状态
接收信息的处理
实时的接收客户端的连接
异常情况的处理包括指示灯的熄灭
开始或重新启动线程
接受到

新连接的登记与处理

```
}
(2) 客户端 SocketClient 的设计与实现
public class TranClient extends Thread {
    private String clientName;
    private String serverIP;
    private int serverPort = 6868;
    private TranClientListener listener;
    private java.net.Socket socket;
    private java.io.InputStream is;
    private java.io.OutputStream os;
    private java.io.ObjectInputStream ois;
    private java.io.ObjectOutputStream oos;
    private boolean isOnline = false;
    public void run() {}
    public void sendMessage(Object obj) {}
    public void socketException(Object source) {}
```

客户端连接名
服务器的 IP 地址
服务器的端口号
客户的监听端口
socket 的客户端
对象输入流
对象的输出流
连接指示
客户端的实时处理线程的连接情况，如果非法退出，根据设定的参数值重新连接
信息的发送，
异常情况的处理包括指示灯的熄灭

4 结束语

在铁路局车流计划调度系统全路范围的推广过程中，运用文中所提出的车站车流报告系统实现了站存车的上报、日（班）计划的下达、运行时刻的上报等功能，使得铁路局与车站之间的数据传输变得准确和快捷，极大地减轻了作业人员的劳动强度。

参考文献：

[1] 邱仲藩. JAVA2 从入门到精通[M]. 北京：电子工业出版社，2003.
[2] Bruce Eckel. Thinking in JAVA[M]. 北京：机械工业出版社，2002.