

基于 WEB 的生产实时监控系统的應用

伍志方 福建漳平电厂(364400)

Abstract

In this paper, the design scheme of real-time web-based production monitoring system is introduced, and the method and key technology of using Java and HTML to manage and examine the real-time production data is also discussed.

Keywords: Web, Java, JDBC, real-time, dynamic, thread

摘要

本文介绍了基于 WEB 的生产实时监控系统的建设方案,用 Java 和 HTML 技术处理、查看生产实时数据的方法及关键技术。

关键词: WEB, Java, JDBC, 实时, 动态, 线程

在工业自动化领域里,基于 WEB 的生产实时监控系統已经成为热门的话题,并正在逐步迈向实施。基于 WEB 的生产实时监控系統,是通过计算机网络,连接各生产现场和控制系統的信息,实现整个企业的监控系统信息的采集、分析、统计、存储,让管理层和全厂各部门都能看到过去只有操作人员才能看到的生产现场的实时状况,通过局域网、广域网、国际互联网,使企业内部、企业之间、坐落在不同地区的企业各部门之间交流生产信息,使出差在外的人员能及时了解到生产情况,实现对生产现场的远程调度、指挥决策。本文根据我们的开发体会,介绍基于 WEB 的生产实时监控系统的实现方案及关键技术。

1 基于 WEB 的生产实时监控系統的基本任务和要求

基于 WEB 的生产实时监控系統提供反应生产现场的实时信息,这些信息来自生产现场的表计、设备及 DCS、PLC、SCDAD 等系統,系統同 MIS 管理网相联,网上计算机通过浏览器可实时查看生产数据,对大量的生产现场数据进行复杂的统计分析,为生产计划、调度指挥、事故分析、辅助决策等提供依据。系統应满足如下基本任务和要求:

- 1) 系統采用动态网页发布数据,客户计算机可通过浏览器直接使用,使系統广泛适用。
- 2) 数据发布及时、准确、可靠,刷新速度快,保持和现场同步。
- 3) 保留历史生产数据。

4) 提供丰富的数据访问工具,如实时动态流程监视图、数据棒图、饼图、实时曲线、历史曲线、现场多媒体监控图像,数据查询、统计、分析等。

5) 为各种基于 WEB 的应用程序提供实时和历史数据接口,使系統的功能可横向扩展。

2 系統组成与实现

采用 B/S 结构模式,使用商用快速以太网,通过计算机网络,将生产现场的各类实时数据、画面和曲线等信息,接入网络服务器中,并以 HTML 文本的形式实时发布,使得用户只需在配备有浏览器的计算机中就可看到生产现场的工况信息。系統体系结构如图 1 所示。

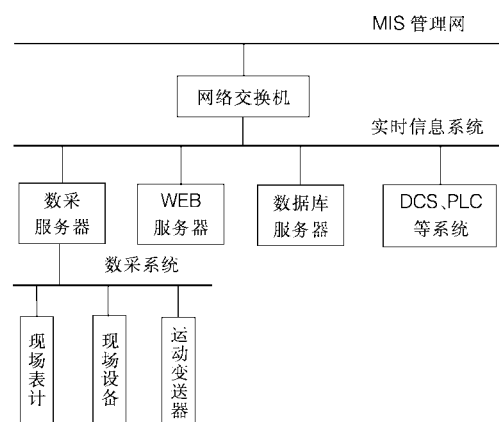


图 1 系統体系结构图

系統采用三层体系结构,最底层为数据采集系統,从现场表计、变送器及其它现场设备中采集数据,

执行控制命令,使用 VC、VB、组态王等软件实现。第二层为实时信息处理系统,现场采集的数据及 DCS、PLC、其它监控系统的数据写入挂接在这一层的 WEB 服务器、数据库服务器,对数据进行统计、分析等处理,将实时数据、历史数据等以 WEB 的形式实时发布。以 WEB 的形式实时发布的信息通过网络交换机等同第三层即 MIS 管理网、广域网、互联网等相连,供用户使用。

WEB 应用由 WEB 页构成,WEB 页是用 HTML 语言编写的网页文件,WEB 网页文件通过浏览器访问,客户计算机通过浏览器可直接访问 WEB 文件,无需安装其他软件,十分方便。但 HTML 语言文件不具备处理实时动态信息的功能,要使 WEB 网页文件能够实时收集、传递、处理、显示实时动态的信息,如实时动态的棒图、饼图、曲线、图像、数据、报表等,我们用 Java 语言对动态数据、图形等进行处理、分析、显示,并把 Java 程序嵌入到 WEB 网页文件中,由网页文件调用。JAVA 语言是一种优秀的动态语言,它编制的程序与平台无关,适用多种系统平台,具有多线程、动态、高性能、可移植、分布式、安全等特点。利用 Java 可编写出动态的、交互的、高效率的基于 WEB 的生产实时监控系統。

3 关键技术

3.1 Java 程序的调用

在 HTML 网页文件中调用 Java 程序的方法是通过一个<APPLET>标记和参数实现的。<APPLET>标记的主要属性有:

CODEBASE 这个属性指定 Java 程序的基 URL;

CODE 这个属性指定要调用的编译后的 Java 程序;

NAME 指定 Java 程序实例的名称;

WIDTH 指定 Java 程序显示区域的宽度;

HEIGHT 指定 Java 程序显示区域的高度。

传递参数的标记为<PARAM>,格式为<PARAM NAME=name VALUE=value>。如调用 Java 程序 bargraph.class 并传递参数 csmc 的程序代码如下:

```
<html>
<head>
<meta name="GENERATOR" Content="Microsoft
FrontPage 4.0">
<title></title>
</head>
<body>
```

```
<p><applet code=bargraph.class height=336
width=755>
<param name="csmc" value="1">
</applet></p>
</body>
</html>
```

3.2 实时信息的传输

生产现场的实时信息数据量大,数据实时变化,刷新周期达秒、毫秒级,要使大量的千变万化的实时数据供网上计算机查询、使用,采用将数据写入数据库,再由 WEB 服务器从数据库中取出数据,再将数据发送给客户计算机的方法显然是不行的,这样的方法不仅容易使磁盘频繁的读写数据造成损坏,也会因为系统繁忙效率低速度慢无法使信息保持和生产现场同步。使用 Winsocket 技术可以很好地解决这一问题。Winsocket 是基于消息的异步存取策略,用于在 Internet 上传输数据和交换信息,在使用中 Winsocket 分为服务端和客户端,服务端和客户端之间可以相互通信进行数据传输,如客户端向服务端发送数据请求,服务端对接收到的请求消息进行处理后发送相应的数据给客户端。我们把数据采集服务器定义为 Winsocket 服务端,客户机定义为客户端,这样就可使现场数据实时地传输给各客户计算机。用 Java 程序实现的代码如下:

服务端的程序:

```
ServerSocket server;
Socket ssocket;
try {
server = new ServerSocket(5000);
svsk = server.accept(); //接收客户端请求
OutputStream out = serverSocket.getOutputStream();
out.write(*Test*);
}
catch (IOException e) {}
```

客户端的程序:

```
Socket csocket;
try {
cSocket = new Socket ("10.35.64.100",5000); //服务器 IP 及端口
out = csocket.getOutputStream();
out.write(send_txt1); //发送读数据请求
in = csocket.getInputStream();
in.read(rec); //读取数据
}
catch (UnknownHostException e) {}
catch (IOException e) {}
```

若数据采集服务器采用 VC、VB 或其他语言编程,则可用相应的语言编写服务端程序。

3.3 数据库的连接

数据的分析、统计,历史数据的保留,需将数据存储在数据库中。企业生产实时数据量大,应使用大型数据库如 Oracle 数据库,DCS、PLC 等监控系统数采前端可根据所采用的工具语言使用 ADO、ODBC 等方法连接数据库,将数据写入数据库。WEB 服务器通过 Java 语言调用数据库,Java 语言连接数据库的方式有 ODBC、JDBC 等,使用 ODBC 连接数据库存在一些问题,如:ODBC 是以 C 语言编写的接口,从 Java 调用 C 语言的原生码会有一些安全、稳定、移植等问题,从 ODBC 的 C 应用程序接口所处理的字符串常量与 Java 的字符串常量转换结果可能不需要,使用 ODBC 必须每台用户端计算机都要安装驱动程序和管理员。使用 JDBC 连接数据库可以解决 ODBC 连接的问题,JDBC 容易使用,自动安装完成,用户端计算机不需要安装程序,并可移植到所有网络、计算机的 Java 工作平台。程序代码如下:

```
public void init(){
    try {          //驱动程序
        Class.forName("oracle.jdbc.driver.OracleDriver");
    }
    catch(ClassNotFoundException ce){}
}
public void paint(Graphics g){
    try {
        con=DriverManager.getConnection("jdbc:oracle:thin:
prdss/prdssuser@10.35.64.100:1521:orcl"); //连接数据库
        stmt=con.createStatement();
        rs=stmt.executeQuery("select dl_0 from zdl ");//查询语句
        while(rs.next()){
            g.drawString(""+rs.getInt("dl_0"),10,20);//读取数据
        }
        stmt.close();
        con.close();
    }
    catch(SQLException e){}
}
```

3.4 使用多线程

多线程是可将程序任务分成几个可以同时进行的子任务,一般的程序只有一个线程,但 Java 程序允许多个线程同时执行。为达到数据的动态刷新,我们在程序中使用了多线程。因为数据、画面的动态显

示是通过重复地刷新显示区域,同时显示的内容产生变化而形成的,要重新刷新显示区域,并不是用无限循环语句不断地进行重复刷新,这样的程序运行时会象死锁一样,再也不能做其它工作了,最好的办法就是定期启动程序刷新显示区域,而线程正好很合适这一工作,它可以用 sleep(time)方法每隔一段时间启动程序刷新显示区域。

3.5 闪烁问题的解决

画面的动态刷新显示,会使屏幕上画面一闪一闪的,如果计算机的速度较慢的话,这一现象就更明显。有效的解决闪烁问题一直是动态画面的关键技术,因为没有人会愿意观看刺眼的一闪一闪的画面。Java 语言是通过调用 repaint()方法通知系统进行重画,repaint()方法实际调用了 update()方法先清洗整个应用程序区域,然后再调用 paint()方法绘制屏幕。很明显,问题就出在 update()方法里,它每次清除屏幕,使画面每次都从有内容,到全空白,再画上内容,无疑造成了闪烁。下面就是 update()方法的缺省代码:

```
public void update(Graphics g){
    g.setColor(getBackground()); //将背景色置为当前绘图颜色
    g.fillRect(0,0,width,height); //用背景色填充整个应用程序区域
    g.setColor(getForeground()); //将当前绘图颜色设回前景色
    paint(g); //进行重画
}
```

很显然,我们必须覆盖这一方法,去改变它每次都呆板的用背景色去填充一下整个应用程序区域。其实,屏幕的显示区域的许多内容是不变的,只有显示动态数据的区域需改变,若只清除改变数据的区域,其它区域用叠加显示,这不会破坏整个画面的内容,却可解决闪烁的问题。下面,就是我们覆盖缺省的 update()方法后的新代码:

```
public void update(Graphics g){
    g.clearRect(x1,y1,x2,y2); //用背景色填充部分区域
    paint(g);
}
```

3.6 使用异常处理技术

在程序执行期间,可能会有许多意外的事件发生。例如,网络通信失败、数据采集异常等。而要一个程序可靠地运行必须进行异常处理,异常,即 Excep-

tion, 是指当 Java 程序违反 Java 语言的语义限制时, 就向程序发出错误信号。在 Java 中使用 try - catch-finally 进行异常处理。我们把可能发生异常情况的代码段放在 try 语句的程序段中, 利用 try 语句对部分代码进行监视, 如果发生异常就进入到 catch 语句中处理。程序代码如下:

```
public void run(){
    try{
        csocket = new Socket("10.35.64.22",5000);
        .....
        repaint();
        runThread.sleep(delay);
    }
    catch(InterruptedException e){showStatus("通讯
错误!");}
    catch (IOException e) { }
}
```

4 结束语

随着 Internet 的飞速发展, 加快了信息的交流和信息的共享, 工业监控系统由传统的 DOS 和 Win-

(上接第 23 页)

所节约的等待确认时间中, 相当一部分是由通信线路的延时造成的。

第二, 由图 1 可以看出, 如果在通信中上行与下行的数据量都很大, 那么使用滑动窗口协议提高传输效率的效果将大大优于仅单向数据量很大的情况。如果上行与下行的数据量都很小, 那么在单片机通信中使用滑动窗口协议的意义将不是很大。

第三, 使用滑动窗口协议必须选择合适的数据分组大小及滑动窗口大小。选择合适分组大小的意义在于避免传输错误时需要重传的单位分组过于庞大以致降低传输效率。选择合适滑动窗口大小的意义在于尽量使等待回应的的时间降到最低并且节约内存资源。必须指出, 当滑动窗口大小为 1 时, 实际上就是传统的应答方式。而选择过大的滑动窗口, 必定会占用过多的内存空间, 并且增加 CPU 的处理负担。

选择的原则主要是考虑所使用的线路的通信质量以及信号延时时间。一般来说, 若所使用线路的通信质量较好, 例如无强烈电磁干扰的 RS232 直接连接或者单位内部程控交换电话网等, 可选择较大的分组。若所使用的线路信号延时较大, 例如长途或者跨区间的市话网等, 则应选择较大的滑动窗口。具体情况最好通过试验来确定。

dows 风格转向 WEB 风格是一种潮流和必然, 借助浏览器, 用户可监控、查询企业的各种生产情况, 基于 WEB 的生产实时监控系统的應用将对企业的生产和发展起到关键的作用。本文介绍的系統已在福建漳平电厂得到运用, 并已通过了验收, 该系统投运后, 使各级领导、职工可方便地查询、掌握生产现场的生产实时信息、历史数据、机组参数、系统潮流等信息, 为各级领导指挥决策、调度生产服务, 为运行部门、检修部门及时调整、检修设备服务, 确保设备安全、经济地运行, 使该厂取得良好的效益, 受到了用户的好评。

参考文献

- 1 Java2 核心技术 (USA)Horstmann,C.S. Cornell.G. 朱志等译.机械工业出版社,2000(11)
- 2 Intranet 从入门到精通. Pat Coleman,Peter Dyson.电子工业出版社,1998
- 3 Steven Clarke. Port Forwarding[EB/OL]. <http://www.ox.compsoc.net/~steve/portforwarding.html>.

[收稿日期:2001.10.19]

第四, 在本文的应用实例中, 只是实现了双机通讯, 实际上, 滑动窗口协议在多机通信中实现也是毫无问题的。

第五, 一般来说, 滑动窗口协议主要适用于全双工通信, 对于半双工通信, 如两线的 RS485 方式等, 实现起来稍显复杂, 对此有兴趣的读者可以与我们联系以作进一步的探讨。

4 意义与展望

在单片机通信中运用滑动窗口技术的意义不仅在于使众多对 TCP/IP 协议不是十分了解的单片机开发人员掌握 TCP/IP 协议中最基本的技术, 从而提高单片机通信的效率, 而且为进一步在单片机通信中根据具体需要真正实现 TCP/IP 协议族中的某些协议打下良好基础。

单片机与互联网的結合是当前单片机或者说嵌入式系统发展的一大方向, 是实现工业化与信息化相结合的重要环节。这一趋势是不可逆转的。

参考文献

- 1 孙涵芳.Intel16 位单片机.北京航空航天大学出版社
- 2 Andrew S.Tanenbaum.计算机网络.清华大学出版社

[收稿日期:2001.8.19]