

33-34,54

Intranet网

数据库

分布式处理

(11)

计算机科学2000Vol. 27No. 8

AR 586

Intranet 数据库:一种分布式处理实现方案

A Solution Implementing Distribute Processing in Intranet DB

王爱清 周以鸣¹ 邹 蓉

(首都师范大学计算机系 北京100037)

TP393.18

Abstract Not same as with CGI or API, a solution with Java can implement distribute processing in Intranet DB system. Java Applet which is often embeded in Web pages can be download by client and run by browser on it. So some job can be done by it to reduce server's load. Meanwhile, Java Application can realize the functions of complex communication. So it can improve efficiency and reduce the transport load.

Keywords Intranet, Java, JDBC, CGI, Distribute processing

企事业组织内建立起自己的内部网,使网络内的各个部门、各种不同平台的计算机可以轻松地相互传递信息;同时通过安全隔离,可以实现与外部的交换传递,这就是如今非常热门的 Intranet。它采用已在世界范围广泛应用的 Internet 技术为它的信息底层服务,其中 WWW 的超文本机制、生动的用户界面、通信实现的方便、灵活,成为 Intranet 上进行信息发布的一种非常流行的趋势,内部网数据库系统的开发实现,现在许多也转向采用这种基于 Web 技术,向使用者提供一种简单、方便、易于操作使用的信息管理方式。

一、Intranet 数据库技术的实现方法

在 Intranet 数据库技术中,最常见、最普遍的解决方案是采用 CGI(公共网关接口)技术。CGI 是一个成熟的并被广泛支持的标准,CGI 规范在 Web 服务器、外部程序和浏览器之间建立了标准的信息交换方式,使得它们之间可以进行信息交流,其工作过程见图1。WWW 服务器收到用户的传输请求后,启动 CGI 外部程序,它将用户数据请求转换为相应的 SQL 语句,由数据库处理执行后,将结果数据再封装为标准的 HTML 文档,WWW 服务器最终返回对用户请求的处理结果。通过使用 CGI 程序连接 Web 服务器和数据库,使得外部程序能动态生成 HTML,因此 CGI 成为基于 Web 的 MIS 的流行方法。

CGI 通过基于文件的接口方式在服务器和外部程序之间传输数据,当多用户请求时,CGI 存在效率不高的问题,首先 Web 服务器是一个瓶颈,浏览器和数据库之间的通信必须经过中间的 Web 服务器和外部程序,所以对每个用户的数据处理请求或每个数据库服

务器的响应,服务器必须实施 HTML 文档和普通数据之间的转换。其次 CGI 应用程序在服务器端运行,相对来说,大量的处理更加重了服务器端的负担;而且用户的每一次请求,都要调用一个 CGI 进程和数据库进行连接、处理,效率相对不高,但 CGI 方式作为一种成熟的技术仍然是很多 Intranet 数据库处理的首选。



图1 基于 CGI 的解决方案

另一种方式是采用 API(应用编程接口)编程方式,如 ISAPI 或 NSAPI 等,由于 API 函数和服务端本身共享进程空间,运行效率提高,API 功能比 CGI 功能快2~5倍。但最大问题是,服务器软件是否能支持 API,相对来说,大多数服务器软件都支持 CGI。

Intranet 数据库实现中的高效、可靠、适合于多种数据库系统的解决方案仍是目前探讨的主要课题,也是正在迅速发展的电子商务需要解决的关键技术。

二、Java 实现 Intranet 数据库:一种分布式处理解决方案

Java 是由 SUN 公司开发的一种面向对象的编程语言,它可以用来编写嵌入 Web 主页的小程序 Applet,在 HTML 文件中,用标记来标识 Applet 程序,如需运行 example.class,代码部分如下:

```

<applet code = example.class width = 500 height =
400>
(PARAM NAME = quantity, VALUE = 5) (/

```

APPLET)

当客户端浏览器读取携带 Java Applet 小程序的 HTML 网页时,Applet 便被下载到客户端,由 Browser 负责 Applet 的解释并执行,其工作过程见图2。Java 这一特性,吸引了很多人在其 Web 页面中用 Java Applet 来增加页面的动态效果。

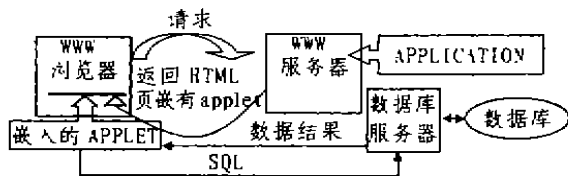


图2 基于 Java 的实现方法

2.1 扩展服务器功能,丰富 Internet 服务

运用 Java 还能够建立独立运行的 Application, Intranet 上还需要很多应用程序来丰富服务,服务器端可以装有 Java 解释器,通过开发强有力的 Java 应用程序,可以扩展服务器的功能和应用,丰富 Internet 服务,例如它可以监听浏览器端的数据处理请求,接受或传递数据,实现特殊的数据通讯过程,满足复杂的通讯要求。总的来说,Applet 的功能不如 Application 强,最主要原因是从安全角度考虑 Applet 有许多限制,如为了防止运行的 Applet 对客户机造成损害,限制 Java 程序在客户机的权限,它不能改写或删除客户机上的文件等。

当然 Application 也可以 C/S 方式作为客户端程序,直接与服务器的 Application 或其它 Internet 服务通讯,包括请求 DBMS 进行数据管理与统计查询,但在 Intranet 系统中客户端采用 Web 浏览器能带来更多的方便。从图2可以看出,Java Application 可以运行在服务器端,扩展服务器功能。

2.2 JDBC API 和 JDBC Driver

JDBC 可以实现 Applet、Application 和数据库之间的通信,它实现了 Java 程序与数据库之间的无缝连接。JDBC 分为两部分:JDBC API 和 JDBC Driver。

JDBC API 可以使编程人员通过统一的接口,访问多种关系型数据库。它由多个类和接口组成,Java Applet 通过 JDBC 技术可以不经 Web 服务器,直接和数据库相连,进行数据的传递,见图2。

JDBC Driver 是 JDBC 驱动器接口,用于连接 JDBC API 和数据库的 API,有多种 JDBC API 需要的接口,正是通过 JDBC Driver, JDBC API 可以按相同的方式和不同的数据库相连,它们之间的关系如图3。通过 JDBC,Java 程序和数据库的通信包括:

- 建立一个连接:JDBC API 中的 java.sql. Con-

nection 类,用于和特定的数据库进行直接连接;

- 执行 SQL 语句:JDBC API 中的 java.sql. Statement 类,支持 SQL 语句的执行;

- 处理结果:JDBC API 中的 java.sql. ResultSet 类,支持对 SQL 语句查询统计结果的访问。

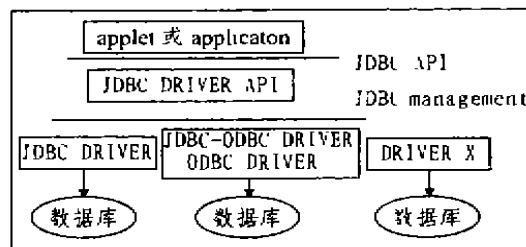


图3 通过 JDBC 和 JDBC 驱动器接口访问数据库

2.3 Java Applet 扩展用户端的处理能力

由于 Applet 在浏览器端运行,客户端通过运行 Applet 可以增强处理能力,同时减轻服务器的负担,例如对用户输入数据有效性的检查,包括数据类型是否正确、是否超出范围,还有是否符合要求,如日期输入中的年份必须输入四位等,如果有错误,不用再将用户数据发送给 HTTP 服务器处理,然后将错误返回,而是客户端自己就可以处理这类检查,这种方式使 Web 更接近于传统的客户机/服务器一类的应用。不仅可以完成检查工作,而且可以对处理的动态数据结果选择一种满意的呈现方式,使其更直观,更一目了然。Java AWT 软件包包括了很多类,通过生成文本框、列表框、选择框等 GUI 对象,可以灵活地组织自己的用户界面,方便用户查询、检索和管理数据。甚至可以选择象图表或报表这样复杂的输出方式处理数据。下面的例子取自教务信息管理系统里的部分代码:

```
public class score_report extends Applet {
    String url = "jdbc:weblogic:mssqlserver4:cptdb
    @202.204.220.12:1433";
    Connection connection;
    ...//其它对象说明
    public void init() {
        ...//用户界面语句
        Class.forName("weblogic.jdbc.mssqlserver4.
        Driver").newInstance(); //连接数据库
        Driver driver1 = new weblogic.jdbc.mssqlserv
        er4.Driver();
        connection = DriverManager.getConnection
        (url,"sa","sq");
        ...
        Statement stmt = connection.createStatement();
        //执行 SQL 语句
        String stsql = "select bjh from t-bjdm";
        ResultSet rs1t = stmt.executeQuery(stsql);
        while(rs1t.next()) { //取得执行结果
```

(下转第54页)

改就可以实现其它支持 SQL 语句的 DBMS 之间数据的转换。

同时 powerbuilder 提供了一种数据转换工具 pipeline, 利用它也可以方便地完成数据的转换任务。

(4) 数据库中表的定义 开发者可以利用数据库服务器提供的数据库完整性约束, 将标准数据库完整性规则的说明作为基表定义的一部分, 即可维护数据参照完整性, 保证了主细目之间级联更新的正确性。

```
Create table person
  id char(6) not null,
  xm char(8) not null,
  :
  constraint primary key(id);
Create table rs_zckh(
  id char(6) not null,
  zclb char(6),
  :
  serino char(8) not null,
  constraint primary key serino,
  constraint foreign key(id) references person(id);
```

(5) 触发器 下面定义触发器 update-person, 主要用于当 person 表发生更新时, 相应的身份表如职称考核、学历教育等也发生变化。

```
create trigger update-person
on person
for update
as
if update(id)
begin
update rs_zckh
set rs_zckh.rybh=inserted.id
from rs_zckh,deleted,inserted
where rs_zckh.rybh=deleted.id
update rs_xlly
set rs_xlly.rybh=inserted.id
from rs_xlly,deleted,inserted
where rs_xlly.rybh=deleted.id
```

(上接第34页)

```
ch_bj.addItem(rsIt.getString(1).trim());
}
stmt.close();
...//其它代码
}
//end init
...
public void paint(Graphics g){
g.setColor(Color.black);
g.drawLine(x0,y0,x1,y0);
g.drawLine(x0,y1,x0,y0);
g.setColor(Color.red);
g.fill3DRect(x0,y0-score[1],40,score[1],
true);
...//直方图的绘制
}
}
```

• 54 •

end

(6) 规则 下面定义中文简拼的规则 RUL-ZWJP, 可以绑定到 PERSON 表的 ZWJP 字段, 作为该字段的值域规则:

```
Create rule rul-zwjp
As (@zwjp like "[A-Z][A-Z][A-Z][A-Z]"
GO
Sp_bindrule rul-zwjp, "person.zwjp"
```

4. 系统的动态集成

我们设计的构件都遵循 PBL 的形态, 同时通过 OLE2.0, DDE 和 DLL 以及 powerbuilder5.0 提供的 library export 和 libraryimport 等函数, 我们可以使市场上 OLE/COM/DCOM/Activex 的构件能与原系统实现无缝连接, 能存取系统中原有的数据, 新的数据库表结构也能通过动态的数据窗口, 通过数据事务对象可以调用原有的构件, 达到构件软件的桌面集成的目的, 这样可以提高应用系统的开发效率, 缩短应用系统的开发周期, 并能提高软件的可重用性。

参考文献

- 董荣胜. 两层和三层 Client/Server 结构的分析. 计算机工程, 1998(6)
- Gogolla M. et al. A Development Environment for an Object Specification Language. IEEE Trans. on knowledge and data engineering, 1995, 7(3): 505~507
- Deubler Hans-Helmut, et al. Introducing object orientation into large and complex system. IEEE Trans. on software engineering, 1994, 20(1): 432~440
- Prieto-Diaz R., Freeman P. Classfying Software for Reusability. IEEE software, 1987, 4(1): 6~16
- 诸葛海. 面向对象的 MTS 开发方法 DDD. 软件学报, 1995, 增刊
- 耿刚复, 李洲明. 基于构件的应用软件系统的体系结构及其开发模型. 计算机研究与发展, 1998, 35(7)

Intranet 数据库中分布式处理的实现方案见图3, 正是因为 Java 能使我们在 Intranet 上能完成复杂、或简单的各式任务, 同时它还可以帮助扩展服务器、客户端的功能, 使服务器端能根据需要建立相应的各种通讯方式, 又可以使客户端具有处理功能, 而不是单纯地依靠服务器, 我们可以采用有效的方式, 使服务器、客户端各司其职, 分工合作, 以减轻服务器的负担, 提高效率, 同时减少不必要的网络数据传输, 提高整个 Intranet 的系统性能。

参考文献

- Mohsen P. Web 数据库开发人员指南. 北京: 机械工业出版社, 1997
- Ablan J. 用 Java 开发 Intranet 应用. 北京: 机械工业出版社, 1997
- Uanhelsuwe L. Java 从入门到精通. 北京: 电子工业出版社, 1997
- 许伟. 传统 MIS 模式向 Intranet 转换. 北京: 电子与信息化, 2000, (3): 31