在分布式计算环境中 Java 的应用

李方敏 彭小兵 (湘潭工学院计算机应用研究所 411201)

摘要:本文通过将基于 CGI 的数据库访问方案与基于 Java 的方案的比较,指出了 Java 将提供高度的灵活性、可伸缩性、可移植性和稳定性。通过使用 HORB——一种基于目标请求代理的软件工具,Java 客户和服务器能容易地创建和透明地访问。然后提出了一种用 Java 和 HORB 开发数据库应用的方法,利用该方法很容易在协同环境建立强健的 Web 应用。

关键词:WWW 目标请求代理 Java applet 协同环境

1. 基于 CGI 的数据库访问

基于 web 的典型数据库应用包括三部分内容:一个 web 浏览器、一个支持 CGI 程序的 HTTP 服务器和一个数据库服务器。一个数据库查询通过从 web 浏览器向 HTTP 服务器发送一个用户请求而被初始化。HTTP Server 接收请求后,使 CGI 程序装配输入数据构成特定 SQL 语句,然后发送 SQL 语句到数据库服务器。数据库服务器返回查询结果给 CGI 程序,然后再通过 HTTP 服务器传递给 web 客户。由于 CGI 是 HTTP 服务器与外部应用接口的事实标准,因此基于 CGI 的数据库访问是当前最通用的方法。

CGI 最初是为 web 的 HTTP 服务器和其他用户应用服务器提供一般的接口而设计,尽管目前 CGI 具有简单性和广泛的应用范围,但基于 CGI 的数据库访问有许多缺点。首先在客户和后台数据库之间的通信总要经过 HTTP 服务器,当有大量用户同时访问 HTTP 服务器时,这将成为瓶颈。对每一被 web 客户提交的请求或数据服务器的每一个响应,HTTP 服务器都必须对数据进行格式转换,这大大增加了查询处理的负载。

第二是基于 CGI 方案缺乏有效性和事务处理支持。对每一个通过 CGI 提交给后台数据库服务器的查询请求,数据库服务器必须完成同样的登录和注销过程,即使对同一个用户提交的连续查询请求也是如此,这种处理当然消耗了大量的时间和资源,尤其当有大量客户同时访问数据库时。因为 CGI 程序仅仅能以批处理方式处理查询请求,它不能支持在线数据库事务处理、带控制功能的交互式多重查询,这个问题是从 HTTP 的无序性继承的,无序性意味着服务器对客户请求的响应与以前的请求和响应无关。象 HTTP 这样的无序协议对单查询类型的应用是合适的,它不适用于象数据库主要关心有效性和事务处理功能的应用。

第三是缺乏用户访问控制。因为 HTTP 的无序性,web 客户的数据库查询请求必须用批处理的方式发送,包括用户标识符和口令。web 客户和 HTTP 服务器之间一般以 ASCII 格式传送数据,因此上述方法是不安全的。基于安全方面的考虑,用户标识和口令常常被嵌在 CGI 程序中去访问数据库,用户访问控制由 HTTP 服务器处理。HTTP 服务器的安全性仅仅处理用户域控制而没有不同优先级的数据库访问控制功能,如:读、写、数据库管理员、数据库所有者等。在一个集成环境,需要使用由数据库服务器提供的已存在的用户访问控制功能,以便在整个集成网络环境遵守同样的安全访问优先级。

第四个缺点是由于 HTML 的限制缺乏表示图形。HTML 文档是静态没有图表和二维、三维示意图。虽然 HTML 的变种 VRML 能处理复杂的三维图形,但它还依赖于 HTTP,不能提供直接的通信接口。HTML 通过生成诸如 GIF 文件可能生成表示图形,但这种方法由于是在 HTTP 服务器,由 CGI 程序生成再传送到 web 客户,因此很慢。客户、服务器计算的一个主要优点是所有用户特定的功能(例查询结果的图形化)都应当在客户端完成。如果 web 客户能完成活动的图形功能这是可能的。如果在 web 客户端采用 Java applets 上述所有难题都迎刃而解。Java 是移动代码的第一次成功的商业实现,Java 程序能被传输到远地机器去执行。Java applets 能使用套接字(Socket)产生它自己的网络连接,无需 HTTP 服务器的介入。

从一个 applet 能直接访问数据库,这种方案不仅能消除在 HTTP 服务器基于 CGI 所产生的瓶颈,而且提供

了面向连接的通信。不同于基于 CGI 所产生的一旦查询完成连接就关闭，applet 一旦连接到数据库，只要 applet 是活跃的或用户还在访问中，就一直保持连接。因此，Java applets 能支持交互式的查询和复杂的数据库事务处理，由于没有 HTTP 服务器所强加的限制和访问控制，用户访问控制能通过数据库服务器的本地安全机制所控制。Java 的 AWT 类有丰富的图形功能，因此 Java applet 能处理复杂的表示图形，在一个真正的面向对象环境实现分布式计算。

2. 基于 Java 数据库访问方案

在 web 应用上使用 Java 的真正潜能在于 Java applets 的可移动性和连接性，Java applet 能运行在支持 Java 浏览器上。用 JAVA 创建数据库客户最容易的方法是构造基于特定平台的客户库函数的包装类。由于安全性的原因，仅仅标准的 java 应用能连接到动态连接库，运行在 web 浏览器上的 java applet 一般没有权限访问本地的任何文件。java applet 访问远地服务器的唯一方法是通过 java 套接字接口。

一般有两种方法构造一个 Java applet 去访问远地数据库。其一是使用两层结构，其所有功能均用 java 实现（图 1）。另一种方法使用三层结构，一个独立的 java 服务器用作网关传递在 applet 和远地数据库服务器之间的请求和响应（图 2）。

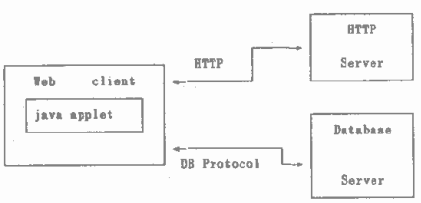

图 1 使用 java 的两层数据库体系结构

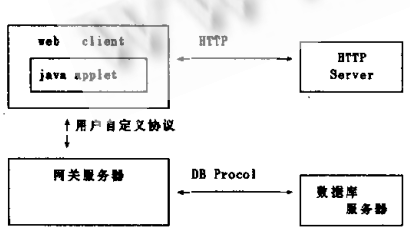

图 2 使用 Java 的三层数据库体系结构

由于 java 客户必须基于特定数据库平台实现，因此前者需大量的编程。而由于网关服务器能同本地封装的 java 类的客户库函数创建作为一个独立的 java 应用，因此后者容易实现。网关服务器一端用用户定义的协议与 java applet 通信，同时另一端用本地的客户/服务器协议与数据库服务器通信。

已经存在的用 java 实现的商业数据库产品倾向于重复 SQL 类型的客户/服务器功能。它们提供大量的 API 类库以便带有参数的大多数数据库函数能被客户设置和发送。结果由于开发者必须熟悉新的 AIPs 去进行数据库访问，限制了开发者的自由。

3. 创建分布式 Java 对象的实验设计

下面通过一个实验说明了使用 HORB 开发数据库应用的方法。我们使用三层体系结构去支持作为 Java applet 实现的数据库客户和 Sybase 数据库服务器之间的交互。用 java 实现的网关服务器既作为 Java applet 的服务器又作为 Sybase 服务器的客户。网关类包含一系列利用 Sybase 数据库函数实现的本地方法，它作为一个独立的 Java 应用运行，通过 HORB 与 apple 通信（图 3）

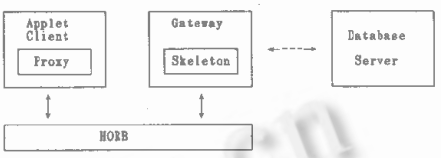

图 3 通过 HORB 的数据库访问

在 HORB 中，一个服务器类文件或一个接口说明能被 HIDL 编译器用来生成服务器框架和客户代理。由于例子较简单，我们选择服务器类文件。网关服务器源代码如下所示：

```
package sybhorb;
class Sybconnect{
        String username;
        String password;
        String sqlquery;
        String fromtable = "from employee where";
    public Sybconnect (Request req, String uname, String passwd){
        username = uname;
```

```
        password = passwd;
        sqlquery = new String("select");
        sqlquery = sqlquery.concat(req.select());
        sqlquery = sqlquery.concat(fromtable);
        sqlquery = sqlquery.concat(req.where());
    }
    public native int execSQL();
    public native void dbLogin();
    public native void dbExit();
    static{System.loadLibrary("Sybjava");}
    {
                ;
public class Server{
  //访问 Sybase 的网关服务器
    public Table makeQuery (Table restable, Request request){
    //用本地 C 库创建接口
    Sybconnect myquery = new Sybconnect (request);
    myuqery.dbLogin();
    myquery.execSQL();
    restable.setTableConnect();
    myquery.dbExit();
    return restable;}
}
```

网关服务器包括类 Sybconnect，该类使用 Sybase 数据库函数实现了本地方法。因为仅仅作为演示目的，只实现三个函数：execSQL()、dbLogin()和 dbExit()。当调用 makeQuery 函数时一个数据库查询被初始化，它接收一个包含查询说明的 Request 对象和存储查询结果的 Table 对象。

在编译服务器类文件并且生成服务器框架和客户代理文件后，用 Java(或 HIDL)编译器能编译客户类，客户源码如下所示：

```
package sybhorb;
import horb.orb.*;
class Client {
    Server - Proxy server;
    Table tbl;
    Client(Table table, Request rqs){
      String host = serverhost;
      //初始化网关服务器
      try{
        HORB orb = new HORB();
      } catch (Exception e) {}
      if (host.length() = = 0) host = "-";
      try{
        Server - Proxy servenew Server - Proxy(new HorbURL(
          host, null));
      } catch (Exception e) {}
      tbl = table;
}
public Table sendQuery (Request rqs) {
      String host = "-";
      Table result = null;
      try {
        result = server.makeQuery(tbl, rqs);
      } catch (Exception e) {}
      return result;
    }
}
```

客户类被定义成独立的 applet 类，它获得存储查询结果的 Table 对象和包含查询请求的 Request 对象中的参数。Table 和 Request 对象在顶层 applet 对象中被定义，直接被链接到用户定义的接口内容。一个被 HIDL 编译器生成的标准的 Server - Proxy 类被 applet 类用来远程引用服务器函数 makeQuery。

使用该体系结构，applet 与网关之间仅需交换一个请求和返回类。apple 可被看作表示商业对象的高级抽象，所有特定的数据库实体封装在网关服务器中，通过在 Sybconnet 类中实现不同的本地数据库函数，客户 applet 能在一个真正的分布式计算环境透明地访问远地数据库。

HORB 的使用消除了处理 Socket 编程的负担，从而在 web 上快速开发分布式应用成为可能。使用 HORB 创建分布式客户和网关服务器的基本处理过程概括成如下步骤：

·为给定的应用定义通用的 Request 和 Table 类

·创建服务器类文件，该类文件含有作为本地方法实现的基于特定平台的库函数的 Sybconnect 类

·用 HIDL 编译服务器类文件生成 Server - Proxy 类

和 Client－Proxy 类

·创建并编译客户类文件

·通过 HORB 运行网关服务器

·启动客户应用(例 applet)访问数据库

4. 结束语

本文说明了用 Java 开发分布数据库应用的一种方法，并将其与传统的基于 CGI 的方案进行了比较。基于对象请求代理的概念，HORB 提供了一种开发 Java C/S 应用的有效方法，在该方法中低级通信被一个请求代理处理，让开发者集中精力于高级应用对象。

参考文献

[1] Arthur van Hoff, Sami Shaio and Orca Starbuck, Hooked on Java(tm), Sun Microsystems Press, 1995

[2] J. Dilley, OODCE: A C++ Framework for the OSF Distriuted Computing Enviroment, Proceedings of the Winter Usenix Conference, January 1995

[3] Hirona Satoshi, HORB, ETL, Japan, http://ring.etl.go.jp/openlab/horb/

(来稿时间：1997 年 9 月)