

Web 服务器与数据库服务器接口技术探析

陈楠 (玉门石油管理局信息中心 735200)

摘要: Web 服务器与数据库的连接技术是基于 Web 的信息管理系统的核心技术。本文详细论述了目前主要的三种接口技术并从 Web 模型入手,对它们的接口原理和开发过程进行了分析与对比,最后对这一接口技术的发展提出了看法。

关键词: 接口 Web 服务器 CGI API JDBC

一、Web 与数据库的集成技术

1. Web 模型

Web 模型是一种从传统的 Client/Server 模型发展起来的新的计算模式。称为 Browser/Server(浏览器/服务器)模型。其本质是三层结构 C/S。C/S 是松散耦合系统,它们通过消息传递机制进行对话,由客户端发出请求给服务器,服务器进行相应的处理后经传递机制送回客户端,其模型见图 1 所示。而 Browser/Server 则把 C/S 模型的服务器端进一步深化,分解成一个应用服务器(Web 服务器)和一个或多个数据库服务器,从而成为三层 C/S 模型,如图 2 所示。

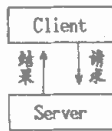


图 1 C/S 模型

在 C/S 模型中,所有的客户端需要配置好几层软件,如操作系统、网络协议软件、客户机软件、开发工具及应用程序等,而在服务器端则是单纯的数据库服务器。Browser/Server 模型则简化了 C/S 的客户端,只需装上操作系统、网络协议软件和浏览器即可,而服务器端则集中了所有的应用逻辑,开发、维护等几乎所有的工作也都集中在服务器端。浏览器从网络上下载应用,Web 的应用服务器将信息组织成分布的超文本,通过超文本标记语言(HTML)和超文本传输协议(HTTP)来描述和组织信息。简便和实用地实现了以整个 Internet 网络为背

景的超文本/超媒体的数据库应用系统。

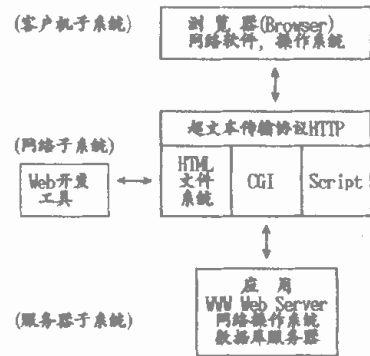


图 2 Web Browser/Server 模型

2. Web 服务器与数据库的连接方法

Web 与数据库的集成并不是什么新技术,早在 Web 诞生的初期,就可以利用通用网关接口(CGI)动态地访问企业数据库。随着应用的普及,Web 服务器与数据库服务器的连接显得越来越重要,许多厂商不断推出新的接口技术,使得这种连接更加简洁、方便和高效。Web 与数据库的接口技术已成为基于 Web 的信息管理系统的核心,为全球贸易奠定了基础。

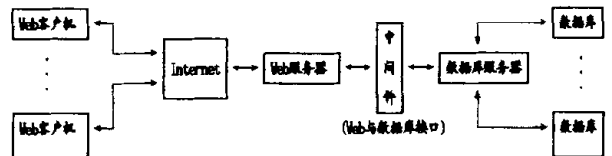


图 3 Web 服务器与数据库的连接方法

二、通用网关接口技术(CGI)

Web浏览器)输入的协同工作数据,完成客户端与服务器的交互操作,扩展服务器的功能,完成服务器本身不能完成的工作。外部程序执行时可以生成 HTML 文档,并将文档返回 Web 服务器。

CGI 应用程序能够与浏览器进行交互作用;还可以通过数据库的 API 与数据库服务器等外部数据源进行通信。如一个 CGI 程序可以从数据库服务器中获取数据,然后格式化为 HTML 文档后发送给浏览器,也可以将从浏览器获得的数据放到数据库中。这样,客户端用户就可以和 Web 服务器进行数据处理。几乎所有的服务器软件都支持 CGI,开发者可以使用任一种 Web 服务器内置语言编写 CGI 语句,其中包括广为流行的 Perl 语言、C、C++、VB、Delphi 等。

按照数据通信方式的不同,CGI 可以分为标准 CGI 和缓冲 CGI。所有的 Web 服务器均应支持标准 CGI,按标准 CGI 编写的程序与具体的 Web 服务器无关。而按缓冲 CGI 编写的程序与具体的 Web 服务器有关。

1. 标准 CGI

标准 CGI 使用命令行参数或环境变量来表示服务器的请求信息,服务器与浏览器间的数据通信采用标准 I/O 方式(如图 4 所示)。

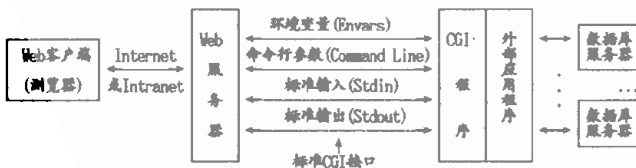


图 4 标准 CGI 接口原理

当客户端的用户完成了一定的输入工作(如填充完 HTML 文档中的 FORM 表)后,向 Web 服务器发出 HTTP 请求(称为 CGI 请求),服务器守护进程接到该请求后,对该请求进行分析,设置所需的环境变量或命令行参数,然后创建一个子进程(称为 CGI 进程)启动 URL 指定的 CGI 程序,在外部 CGI 程序与服务器间数据 I/O 通道(标准 I/O),并与该子进程保持同步,以监测 CGI 程序的运行状态。子进程通过标准输出将执行结果返回给服务器守护进程,守护进程再将处理结果作为应答消息回到客户端。

外部 CGI 程序通过环境变量、命令行参数、标准输

入输出与 Web 服务器进行通信,传递有关的参数和处理结果。CGI 的输出可以有多种类型,如 HTML 文档、图象、纯文本或声音等。它还可以输出指向其他文档的链接。

2. 缓冲 CGI

由于许多 Windows 环境下的编程工具不支持标准 I/O 方式(如 VB 和 Delphi 等),这时就不能用它们来开发基于标准 CGI 的应用程序。

于是提出了缓冲 CGI 的概念。缓冲 CGI 也称为间接 CGI 或 Win CGI。此时,CGI 扩展程序与服务器间通过缓冲 CGI(称为缓冲区)而不是标准的 CGI 接口进行通信,但缓冲 CGI 与服务器间的通信还是通过标准 CGI 接口,由 Web 服务器的内置缓冲处理程序实现(如图 5 所示)。

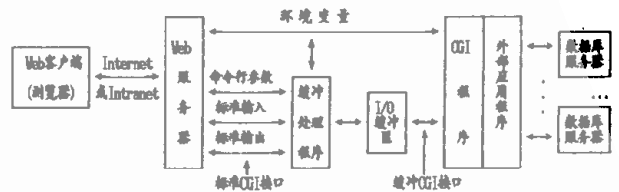


图 5 缓冲 CGI 接口原理

三、Web 服务器应用编程接口(API)

有些 Web 服务器软件厂商针对 CGI 运行效率低下、编程困难等缺点,开发了各自的服务器 API(Application Programming Interface),试图克服 CGI 性能方面的缺陷。

下面以 Microsoft 和 Netscape 公司的产品为例,分析 API 原理。

由于开发 API 应用程序存在许多潜在的困难,所以 Microsoft 与 Netscape 在其各自的服务器中提供了基于 API 的编程接口。Microsoft 提供了 IDC(Internet Database Connector)。Netscape 提供了 LiveWire/LiveWire Pro。

1. Microsoft 的 IDC

IDC 模块是 IIS 的一个动态连接库(DLL),它通过 ODBC(开放数据库联接)访问各类数据库,它使得 Web 页面中包含有数据库信息。可以由用户在页面上输入信息来插入、修改和删除数据库中的信息,允许由 SQL 命令来操作数据库。图 6 是通过 IDC 访问数据库的示意图,这是 Web 服务器端的结构,其中的 HttpODBC.DLL 就是 IDC 模块。

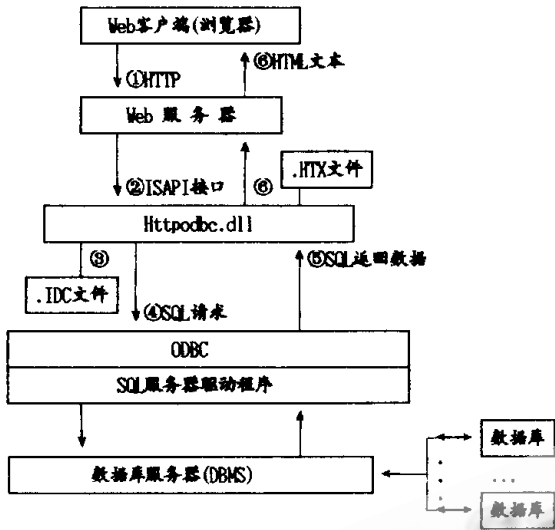


图6 通过 IDC 访问数据库

IDC 使用了两类文件：IDC 脚本文件 (.IDC) 和 HTML 模板文件 (.HTX)。

IDC 的工作原理是：当 Web 服务器 IIS 对由浏览器送过来的 URL 字符串做出反应时，如果该字符串以 .IDC 结束，IIS 就知道这是个 IDC 请求，将其传给 IDC 模块，IDC 依次读取脚本文件并同数据库服务器交互。IDC 模块从数据库得到查询结果后，就用模板文件 (.HTX) 格式化该结果，得到一个实际的 HTML 文档，然后将该文档传送给 Web 服务器 IIS，由 IIS 负责将 HTML 文档返回给浏览器 (依图中序号顺序完成)。

2. Netscape 的 LiveWire/LiveWire Pro

Netscape 的 LiveWire 是一个强大易用的 Web 开发环境，而不仅仅是一个数据库访问接口。LiveWire 是专门设计的全面开发方案，利用该方案开发人员可以开发、管理 Web 页面、Web 网点，并且可以利用 SQL 语句或 ODBC 直接访问数据库。

LiveWire 的开发语言是 JavaScript。在 JavaScript 语言中包含有一个 Database 对象，可以用该对象来打开或关闭数据库连接、查询或修改数据库。例如，可以用 Database.Cursor 来查询数据库，它的参数是一条 SQL 语句；用 Database.Close 关闭与数据库的连接。

LiveWire 的数据库访问函数库是一组用 JavaScript 编写的函数。这些函数支持对 Informix、Sybase、Oracle 的访问，也可以通过 ODBC 访问其他的数据库，如 Microsoft Access、Borland Paradox 等。

数据库访问函数库可以传送 SQL 语句到数据库服务器，由数据库服务器执行完后送回结果，使用透明 (Pass-through) SQL 方式。该函数库亦可以数据库中的存储过程，支持数据库的触发器，支持事务处理 (Transaction)。

同 Microsoft 的 IDC 一样，LiveWire 也仅支持自己的 Web 服务器，对其他厂商的服务器软件不予支持。

四、Java 数据库互联接口 (JDBC)

Java 的推出，使 Web 页面有了活力和动感。用户可以从 Web 服务器上下载 Java 小程序到本地浏览器上运行。这些下载的小程序可以就向本地程序一样，可独立地访问本地和其他服务器的资源。

1. JDBC 原理

JDBC 是一种可用于执行 SQL 语句的数据库 API，它由一些 Java 语言写的类、界面组成。它在功能上与 ODBC 相同，给开发人员提供了一个统一的、标准的数据库访问接口。

通过使用 JDBC，开发人员可以方便地将 SQL 语句传送给几乎任何一种数据库。用 Java 编写的程序能够自动地将 SQL 语句传送给相应的数据库服务器。而且，Java 语言的平台无关性使 Java 与 JDBC 的结合所开发的数据库应用系统真正实现 “Write Once, Run Everywhere!”

JDBC 访问数据库的原理是：Internet/Intranet 用户使用浏览器连接到本地或异地的 Web 服务器上，下载含有 Java 小程序的 HTML 文档。如果 Java 小程序使用了 JDBC，浏览器运行的小程序可直接与数据库建立连接 (如图 7 所示)。

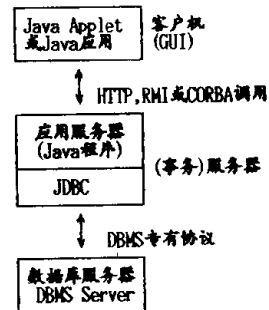


图7 通过 JDBC 访问数据库

2. JDBC 实现

JDBC API 是通过 Java 的类接口实现的，几个主要

的接口是:

①java.sql.DriverManager: 负责处理JDBC数据库驱动程序加载和卸载。Drivermanager类作为JDBC的管理层,负责找到并装入与URL指定的数据库相符的JDBC驱动程序。该驱动程序与远程数据库相连,返回一个java.sql.connection对象。

②java.sql.Connection: 负责建立与数据库的连接。

③java.sql.ResultSet: 表示从数据库服务器返回的结果集。通过操作给结果集实现对数据的访问。

下面的Java程序段将连接一个ODBC数据源并执行一条SQL语句,返回的结果集放在结果集对象RS中。

```
String url = "jdbc:odbc:Factory";
Connection con = DriverManager.getConnection(url, "UserName", "PassWord");
Statement stmt = con.createStatement();
ResultSet RS = stmt.executeQuery("SELECT Eno, Ename, Esex FROM Clerk");
```

JDBC数据库采用URL的指名规则。JDBCURL结构为:Jdbc:<SubProtocol>:<SubName>,其中SubProtocol表示一个特定的数据库连接,SubName的内容依赖于SubProtocol,"jdbc:odbc:Factory"指定一个ODBC数据源Factory。

五、各种接口技术的比较

虽然CGI、Web Server API和JDBC都能实现Web与数据库的连接,但它们的工作原理是不同的:

CGI和API是将Web服务器与某一进程或API相连,该进程或API将得到的结果格式化为HTML文档,返回给Web服务器,然后由Web服务器将得到的HTML文档传给浏览器。就是说,与数据库的交互是由Web服务器完成的。而在JDBC中,浏览器将嵌于HTML文档中的Java应用程序下载到能够运行Java程序的本地浏览器中加以运行。就是说,与数据库服务器的交互是由浏览器直接完成的。

表1对它们从不同侧面进行了比较。

相比较而言,CGI接口应用灵活,开发工具丰富,功能范围广,技术相对成熟。CGI是一个大家都支持的规范,但是用CGI编程太复杂,一个简单的Select查询,用CGI来实现就需要上百条语句;CGI运行效率也不高,并且要求程序员必须熟悉HTML。再加上与平台有关的缺点,因而CGI会被新的规范所取代。

用Web服务器API编写的程序运行效率高,但是,用API编程比CGI还困难,对程序员的要求更高。并且

API致命的弱点是,各API间没有互通性,用一种API编写的程序不能到另一种Web服务器上运行,限制了使用范围。尽管个别公司(如Oracle)推出了基于API的开发工具,在一定程度上解决了API开发困难的问题,但仍存在与API一样的弱点:与各自Web服务器捆绑的太死,缺乏通用性。

表1 各种接口技术比较

	CGI	API	JDBC	IDC
实现的灵活性	好	好	好	好
编程的复杂度	复杂	复杂	中等	简单
对程序员的要求	高	高	中等	低
开发时间	长	长	中等	短
可移植性	较好	差	好	差
CPU负担	重	较轻	较低	较低
平台有关性	有关	有关	无关	有关

Java编程规范,接口清晰,开发效率高。Java代码在可移植性、安全性和可靠性等方面也有显著优势,利用Java的多线程控制技术可以大大提高程序的执行效率。而且Java翻译器是在客户端运行,开发人员能创建反馈迅速的Java Applet,这是CGI和API编程无法实现的。虽然Java缺乏组件结构,交叉平台的GUI构架很粗糙,技术还不够成熟,功能也有待扩展。但JDBC仍是最有前途的接口技术。JDBC借鉴了ODBC的思想,可以利用现成的ODBC驱动程序访问各类数据库,保护了原有的投资。重要的是,在数据库方面,所有的数据库厂商都宣布支持JDBC标准,已推出或准备推出各自的JDBC驱动程序;在Web服务器方面,除了Microsoft坚持自己的ActiveX外,其他厂商都宣布支持JDBC标准。随着时间的推移, JDBC会取代CGI成为新的通用的接口规范。

参考文献

- [1] Jerry Ablan, Developing Intranet Applications with Java, Sams.net. 1996.
- [2] Michael Dieckmann, When Worlds Collide: The Internet Meets the Corporate Database, in Database Programming&Design Aug. 1995, pp48-54.
- [3] T. Berners-lee, "Hypertext Transfer Protocol", Internet Engineering Task Force Working Draft, November 1993.

(来稿时间:1997年11月)