维普资讯 http://www.cqvip.com

计算机科学 1999Vol. 26№. 6

基于CORBA的Web数据库发布技术*)

CORBA-Based Database Publishing

薛 慧 代亚非

(哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

Abstract On the research work of CORBA technology, this paper proposes a new database publishing method: the database publishing based on CORBA. The paper outlines CORBA firstly, then puts forward the structure of the database publishing model based on CORBA, at last the concrete scheme of this structure model is proposed.

Keywords Database publishing, Web, CORBA, Client/Server

1 引言

数据库技术经过几十年的发展,其功能越来越强大,但随着以网络为中心的计算机时代的到来,以往被广泛使用的用传统DBMS建造的数据库资源不能被Web直接访问,这种状况影响了数据库资源的共享,造成了资源浪费。而Web上数据量大、类型多,管理不足,而这恰恰是数据库的优势,因此数据库技术与Web技术的结合已成为当前研究的热点。

我们这里所提到的"数据库发布"是指在以Client/Server为工作模式的Web环境下,在Server端提供数据库信息,在Client端用Web浏览器查询数据库的内容。数据库发布从发布者的角度来看要尽量简单;而从适用范围来看,又要求整个发布系统具有开放性和平台独立性。近两年来人们提出了多种通过Web访问数据库的方法和途径,总体来看可以分为三种:CGI方式、Web API方式、JDBC方式。CGI作为最早的数据库与Web的连接方式,得到了广泛的应用,但是采用CGI方式执行效率低,为了克服CGI的局限,一些Web服务器厂商开发出专用的API,允许开发人员编写程序来扩展服务器的功能,这就是Web API技术,如Netscape提供的NSAPI,Microsoft提供的ISAPI等。Web API方式是采用一种线性编程方式,因此效率比CGI高,而且可以使用Web服务器本身的安全机制,但它依赖于特定的服务器。随着Java的盛行,第一个与DBMS连接的标准草案完成,这就是Sun公司推出的JDBC。采用JDBC技术实现数据库与Web的连接,充分利用了Java语言的优点,因此具有很好的健壮性和灵活性,但这一技术尚未成熟,仍在发展之中。

尽管已经有一些相关的工作在进行,如CGI、Web API、JDBC等方式,但是现有的这些方法都要求数据库发布者具有良好的编程基础,会用相应的工具来开发数据库访问的应用程序,这些要求无疑限制了大多数人的发布愿望。因此,提供一种简单有效的数据库发布系统,以免除数据库发布者的编程环节,不仅会进一步扩大Web技术的应用面,而且有益于数据库信息的共享。CORBA技术的兴起,给Web的发展带来了新的契机。CORBA具有跨平台的互操作性、可扩展性和业界广泛的支持,特别适合构造灵活的客户机/服务器系统。为了使数据库发布简单又能够跨平台,在研究了CORBA体系结构的基础上,我们将CORBA技术应用在数据库发布中。

2 基于CORBA的数据库发布模型

2.1 CORBA概述

CORBA是Common Object Request Broker Architecture(通用对象请求代理结构)的缩写,它是OMG(对象管理组)于1991年发布的构造分布式系统结构的技术规范。它定义了分布式对象如何实现互操作,目的是使基于对象的软件部件在分布异构环境下可重用、可移植和可互操作。CORBA规范定义了IDL语言及映射,ORB体系结构和ORB间互操作机制。CORBA规范具有以下特点:

· 互操作性:CORBA在客户和服务器之间提

*)本文研究由国家自然科学基金资助,批准号为69773040。

维普资讯 http://www.cqvip.com

供了两层可操作性。一方面它将客户与服务器都抽象为对象，所有功能都封装在对象内部，对外提供简单的接口，可被其它对象以动态或静态方式调用。另一方面，对象间的通信是通过 ORB 代理实现，对象不必关心细节，由 ORB 定址发送请求，是 ORB 对象间的"通信总线"。

·提供了软总线机制：所谓软总线是指 CORBA 规范定义了一组接口规范，任何应用程序，软件系统或工具只要具有与该接口规范相符合的接口定义，就能方便地集成到 CORBA 系统中，而这个接口规范是独立于任何实现语言和环境的。不同的对象可以跨越网络和操作系统生存在这个总线上，而这个总线则为对象之间的通讯和交互提供各种服务。通过这个总线，任何被集成到 CORBA 系统中的成员间都可以使用对方提供的所有服务。

·跨平台性：由于 CORBA 规范中定义了 IDL 语言这一与平台无关的语言，并引入代理(ORB)概念，因此 CORBA 应用具有较好的跨平台性。

2.2 基于 CORBA 的数据库发布模型

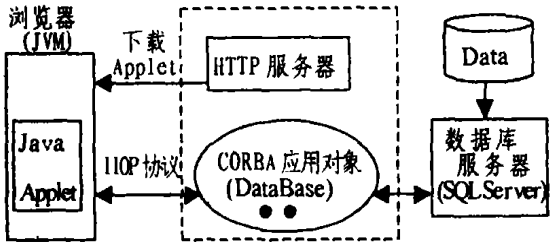

图 1 基于 CORBA 的数据库发布结构

Web 上的数据库发布，最关键的是建立 Web 客户与数据库服务器之间的通信接口，通信接口的不同实现技术形成了各种各样的数据库发布方法。在 CGI 方式中，以 CGI 实现通信接口；在 Web API 方式中，以 Web 服务器 API 实现通信接口；在 JDBC 方式中，以 JDBC 驱动实现通信接口。通过对目前比较流行的分布式对象体系结构 CORBA 的研究，我们利用 CORBA 来实现这个通信接口，提出如图 1 所示的结构模型，以免除发布者的编程环节，实现平台独立性。

在这里，一个用 CORBA 实现的应用对象(DataBase)架起了 Web 客户与数据库服务器之间的"桥梁"。该对象负责与数据库服务器交互，它是脱离 HTTP 服务器而存在的，以一个独立的服务器形式与 HTTP 服务器一起运行，而且该对象服务还可以为其它 CORBA 对象使用。由于 Java 和 CORBA 是互补的两项技术，CORBA 的强健性可以提高 Java 编程语言的性能，我们用 Java Applet 实现数据库访问前端，前端与后端通过 CORBA 的 IIOP 协议进行通信，这对用户来说是透明的。其中 Applet 和 CORBA 对象服务(DataBase)放在同一台机器上，而数据库及其服务器可放在任何可通过网络访问得到的地方。

2.3 后端 CORBA 应用对象的设计

2.3.1 对象接口的定义 任何一个 CORBA 对象的设计都必须用 IDL 定义接口。IDL 使得所有 CORBA 对象以一种方式被描述，仅仅需要一个由本地语言(C/C++、Java 等)到 IDL 的"桥梁"即 IDL 编译器。我们用 IDL 将对象的接口定义如下：

```
//DataBase.idl
module myModule
{
  interface DataBase
    {
    string GetDBInfo(in string dbName,in string URL);
    string  GetDBQuery (in  string  dbQuery, in  string
          URL);
    };
};
```

在这里，为方便描述，我们将该 CORBA 对象命名为 DataBase，对象 DataBase 接口的定义要充分考虑前端的交互需求。由于整个发布系统要使得数据库发布者勿需编程，这就要求前端具有数据库独立性，能适应不同结构的数据库发布。为此，后端需提供获得库结构信息的服务，我们将这一服务定义为 GetDBInfo。另外，后端的查询服务是必不可少的，将其定义为 GetDBQuery。这两个功能是后端需要实现的两个最基本的服务。当然，若需要其它服务，可再添加。

用 IDL 定义一个接口后，接口定义文件作为 IDL 编译器的输入，经编译后，输出客户 Stubs 和实现 Skeletons。客户 Stubs 和实现 Skeletons 与 ORB 配合自动完成如下功能：对象注册和调用，对象定位和连接以及参数的传递等。

2.3.2 对象 DataBase 接口功能的实现 CORBA 对象可以用任何一种 CORBA 软件开发商所支持的语言，如 C、C++、Java 等来编写。同样地，CORBA 对象可以运行在任何一种 CORBA 软件开发商所支持的平台上，如 Solaris、Windows 95/NT、Unix、或 AIX 等。这意味着，我们可以在 Windows 95 下运行 Java 应用程序，同时动态调入并使用 C++对象，而实际上，该对象可能存储于一个在 Internet 上的 Unix Web 服务器上。考虑到要用 SQL 提供给 C 语言的 API，我们用 C 语言编写接口的实现。拟采用 IONA 的 Orbix，Orbix 是用 C 语言写的 CORBA ORB。根据 CORBA 规范，编写对象实现的代码(DataBaseImpl.c)。

```
//将实现 Skeleton 包含进来
String GetDBInfo(CORBA.string dbName,CORBA.string
      URL)
    {
```

维普资讯 http://www.cqvip.com

```
        ...
          元信息处理;
        ...
    }
String GetDBQuery (CORBA. string dbQuery, CORBA.
       string URL)
    {
        ...
          查询信息处理;
        ...
}
```

用 CORBA IDL 可以把任意语言写成的现有程序封装到对象界面中,这也是 CORBA 的一大优点。

2.3.3 *编写对象 DastaBase 的服务* 对象接口的实现只是提供功能的实现模块,类似一个个零件,要提供一个完整的服务,还要编写对象服务程序,它调用对象接口来完成具体的功能。如下是对象服务程序的部分代码:

```
void main(String[]args)
{
      ...
      //初始化 ORB
      CORBA. ORB orb=CORBA. ORB. init();
      //初始化 Basic Object Adapter
      CORBA. BOA boa=orb. BOA_init();
      //创建对象 DataBaseImpl
      DataBaseImpl data_base_impl = new DataBaseImpl
                  (args[0]);
      //引出对象指针
      boa. obj_is_ready(data_base_impl);
      //等待请求
      boa. impl_is_ready();
      ...
}
```

在编写对象服务程序时,首先要对 CORBA 环境初始化,然后创建对象的一个实现,并将对象实现在接口池中注册(引出对象指针),最后通知 ORB 服务就绪。将上述文件编译成可执行文件,并在服务器机上启动运行,随时接受客户(前端)的请求。

2.4 前端的设计

所谓的数据库发布前端是指与最终用户直接打交道的部分,其外层主要完成界面表示,内层完成请求服务以及对请求结果的加工处理。前端通过管理与用户的所有交互,把服务器和网络对用户隐蔽起来,仿佛整个应用只是在局部执行,而没有使用其他的进程、机器和网络。

2.4.1 *前端 applet 的实现方案* 为了使浏览器集成数据库访问前端的功能,我们用 Java applet 来实现数据库访问界面。因此,浏览器必需支持 CORBA 访问,以使 Java applet 能在浏览器中访问 CORBA 对象。我们选用了 Visigenic 的 VisiBroker for Java,这种 ORB 已经经过 Oracle、Netscape 和 Novell 等公司的认证,并已被纳入 Netscape Navigator 4.0。数据库发布从发布者的角度考虑,要求它具有通用性,即对不同的发布者都适用。从用户角度考虑,又要求它交互性强,使用方便。我们在设计前端时充分考虑了这两方面的要求。用户对数据库的访问操作是直接在前端进行的,因此前端要尽量考虑到用户的各种需求。applet 界面构件应包括数据库表选择框、域选择框、条件编辑框、数据库地址输入框以及结果显示窗等。

2.4.2 *将 applet 嵌入主页* 将上述 Java applet 源文件编译成.class 文件,编写 HTML 页面(DatabasePublishing.html),嵌入 Java applet。实际上,这个嵌入了 applet 的 HTML 页面就是用户所看到的前端表现形式。

3 数据库发布的运行模型

对数据库发布者来说,要发布的数据库及其数据库服务器可以与前端 DatabasePublish.html 和后端(即 CORBA 对象服务 DataBase Server)放在同一台机器上;也可以放在不同的机器上(如图 2)。

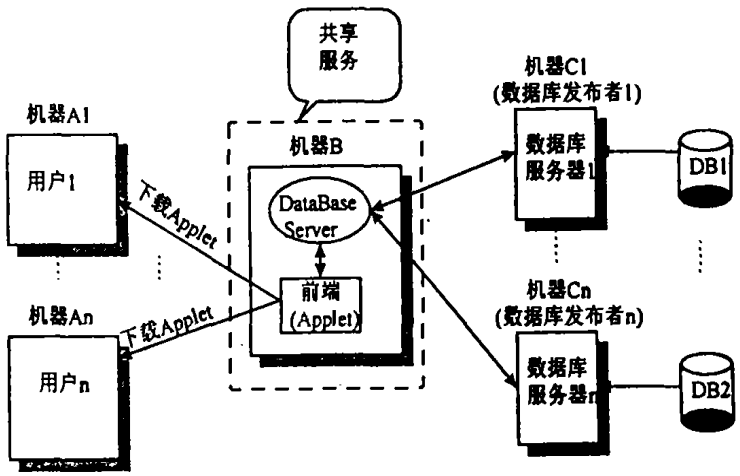

图 2 基于 CORBA 的数据库发布系统运行模型

维普资讯 http://www.cqvip.com

Web 客户只需输入前端所在地址(如 http://pact518/DatabasePublish.html)便可访问数据库信息。其运行过程是这样的:

1)浏览器获得用户的 URL 请求,通过 HTTP 协议将前端 HTML 页面下载到本地执行。

2)applet 从数据库名输入域获得用户键入的数据库名后,通过 ORB 及 CORBA/IIOP 协议调用后端 DataBase Server 的服务 GetDBInfo。

3)DataBase Server 通过与 SQL Server 交互获得数据库结构信息并返回给 applet,applet 将库结构内容显示在表选择框和域选择框中。

(4)用户根据 applet 提供的信息进行查询命令编辑,然后提交数据库查询请求。

5)applet 调用后端 DataBase Server 提供的服务 GetDBResult,DataBase Server 通过与 SQL Server 交互获得查询结果,并返回 applet,applet 将结果加以显示。

结论 本文主要探讨了基于 CORBA 的数据库发布方法,这种发布方法对数据库发布者来说只需给出数据库地址,免除了发布者编程麻烦。由于应用了 CORBA 和 Java,我们所提供的对象服务 DataBase Server 是生存在 CORBA 总线上的,因此整个数据库发布系统可在多种平台上移植,并且可被其他的 CORBA 对象调用,具有开放性和可重用性。其次,DataBase Server 是独立于服务器的,因而对 DataBase Server 的修改与升级并不影响 HTTP 服务器的运行。由于采用了面向对象的设计思想,增加一个服务功能,如增加一个数据库口令服务,只需增加一个接口,因而有良好的可扩充性。

由于 CORBA 的产生时间并不长,各项技术都在发展成熟中,基于 CORBA 的数据库发布还有待于进一步深入研究和开发,并有着广泛的应用前景。

参考文献

1 Omnipotent Corporations. CORBA Tutorial. Available at: http: // www. omnipotent. com/addvalue/corba. html

2 Object Management Group(OMG). The Common Object Request Broker: Architecture and Specification (CORBA),1995

3 刘东波. Web 数据库技术进展. 中国计算机用户,1997(10,上):45~49

4 The NSAPI versus the CGI Interface. Available at: http: // home. de. netscape. com/newref/std/nsapi_vs_cgi. html

5 王克宏,丁锂,孙元. Java 语言 SQL 接口——JDBC 编程技术. 北京:清华大学出版社,1997

(上接第 74 页)

06/30 之间"一类范围的过滤即被称为切块。必须仔细地分析各数据维的可操作性以及固有的相关程度,有时统计图类型(折线图、条形图、堆积图、饼图以及散点图等)也不得不作为一个维来对待。

3 数据挖掘与决策支持系统

数据挖掘是一种从大型数据库或数据仓库中提取隐藏的预测性信息的新技术,它能挖掘出数据间潜在的模式,找出最有价值的信息。数据挖掘的思想依赖于以下三个技术的支持程度:大规模数据采集、功能强大的并行多处理器计算机、数据挖掘算法。

数据挖掘可实现的功能有:自动预测趋势和行为;自动发现以前未知的模式。

虽然数据仓库、OLAP 和数据挖掘最初是作为三种独立的信息处理技术出现的,但是,由于它们内在的联系和互补性,今天,DW+OLAP+DM 的结构已被业内人士公认为是 DSS 的有效解决方案。这种全新的 DSS 构架的真正意义在于重新展示了信息的本质,表明了信息系统的设计观念从处理驱动到数据驱动的转变。在未来的时代,信息的重点将转移到数据模式分析,处理技术则是应数据分析的需要而产生的。

参考文献

1 Hofland P, Utsler J. Data Mining at Your Desk. 1997. 7

2 王珊. 数据仓库、联机分析处理、数据挖掘-基于数据库技术的 DSS 解决方案. 计算机世界,1997(1)

3 Watterson K. Attention, Data-Mart Shoppers, 1997. 7