

41-44

数据库发布系统

web

Internet网

数据库

12

计算机科学1999V:1. 26N:12

# 基于 Web 的数据库发布系统的分析和实现\*

Analysis and Implementation of Web-based Database Publishing System

杜跃进 胡铭增 方滨兴

(哈尔滨工业大学计算机系 哈尔滨 150001)

TP393

TP311.13

**Abstract** Accessing database via Web is quite popular a research field nowadays. But in fact the concept of Database Publishing has its special meaning. The paper distinguishes the differences between Web-based database publishing and other conditions of Web-based database accessing. On the basic of providing the concept of publishing, it discusses the model of Web-based database publishing. It also introduces a system that implements the model. The conclusions are valuable for other Web applications and developments.

**Keywords** Internet, WWW, Database publishing, Web-based applications

## 1 引言

Web 和数据库技术的结合已经成为必然的发展趋势<sup>[1]</sup>,其原因主要来自这几个方面:第一,信息的飞速膨胀使得原来的以 HTML 文件为基础的组织机制遇到巨大挑战,大量的文件使得 Web 信息的管理、更新、查询等变得越来越低效,需要引进数据库技术以改善这种现状;第二,基于 Web 的应用和服务种类越来越多,一般都需要数据库的支持;第三,传统数据库中积累多年的数据逐渐成为 Web 重要的信息源:拥有和正在使用这些数据库的部门和企业需要将其其中的一些数据库公布出来以提供更好的服务和适应新的运行模式;Web 本身也需要更多有价值的数据库来建设和丰富自己。

上述第三种情况和前两种情况有所不同,前两种情况的需求可以通过 Web 或者数据库技术本身的更新或改动来满足,例如扩展 HTML 语言和改动 HTTP 协议使之适合于数据库操作,或者改动数据库和数据库管理系统使之支持 Web 中的超链机制等,但是第三种情况中,首先,不同的是已经存在大量的传统数据库,必须考虑如何可以通过 Web 访问这些传统数据库,而不是如何构造新的可以方便地由 Web 访问的数据库;其次,传统的 MIS 系统在相当长的时间里并不能被 Web-DB 所取代,因此 MIS 系统将管理维护这些内部数据库,只是将需要公布出来的数据库提

交给供外部访问的接口。

本文研究的内容属于第三种情况,即称之为数据库发布。在一个数据库发布系统中,数据库从逻辑上说有私有数据库和公开数据库两种,参与系统的人有数据库提交者、数据库使用者、数据库管理者三种。如果把数据库抽象为一般意义上的资源,把数据库发布抽象为通用的发布概念,则发布行为可以定义为一个六元组:

$$P = (M, p, \delta, S, \xi, R)$$

其中:  $M$  是发布主体的集合,用以确定发布的对象,  $M = \{m_i | i = 1, 2, \dots\}$ ;  $p$  是被发布的对象的集合,  $p = \{p_i | i = 1, 2, \dots\}$ ;  $S$  是传播被发布对象的公共环境的集合,  $S = \{s_i | i = 1, 2, \dots\}$ ;  $R$  是发布受体的集合,  $R = \{r_i | i = 1, 2, \dots\}$ ;  $\xi$  是个函数,决定了发布者将被发布对象传递给发布环境的过程;  $\xi: M \times p \rightarrow S$ ;  $\delta$  是个函数,决定了将被发布对象转交发布受体的过程;  $\delta: S \times p \rightarrow R$ 。

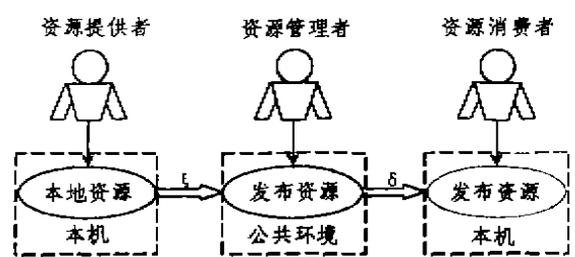


图1 资源发布的角色关系

\* 863计划和国家自然科学基金资助课题(批准号:69773040),杜跃进 博士生,研究方向为基于 Web 的协作计算,胡铭增 博士生导师,研究方向为体系结构,并行处理等,方滨兴 博士导师,研究方向为计算机网络,高性能计算机体系结构。

发布行为的核心是发布的资源(对象),它把牵涉到发布行为中的几种角色联系在一起,如图1所示,资源的提供者创造和公布资源;资源的消费者寻找和使用发布的资源;发布资源的管理者维护这种行为的正常进行,这是发布行为的一个显著特征。

生活中的出版行为就是一个发布的实例,例如书籍和音像制品(p)的出版发行中,M是出版商,S是书市,R是消费者集合, $\xi$ 是出版印刷和上市的过程, $\delta$ 是销售给客户的过程。

不难想象,在发布行为中,发布环境 S 是非常重要的,它决定了被发布的资源是否拥有更大的机会被资源消费者索取。传统的 WWW 系统本身虽然是超媒体信息的发布系统(其发布对象 p 是超媒体文件,发布主体 M 是 Web 管理员或者权限允许的用户,发布环境 S 是构建在 Internet 上的 WWW 服务器、浏览器和它们之间的协议 HTTP),但是由于它具有非常好的普及性、易用性以及其它一些有利条件,使得人们纷纷把它作为其他一些对象的发布环境。例如基于 Web 的计算

资源发布中,S 是 WWW 环境,发布对象 p 是组件、应用程序、或者硬件资源<sup>[1]</sup>。因此,本文讨论的数据库发布系统也采用 WWW(Web)环境作为发布环境 S。

## 2 基于 Web 的数据库发布系统模型

基于 Web 的数据库发布框架模型如图2所示。从这个模型中可以看出数据库发布和 Web-DB 以及 MIS 的不同。在数据库发布的行为模式中,从逻辑上说后端数据库 DB 和 DB'是不同的,DB 是处于企业内部网络中的 MIS 系统的 DBMS 所管理的传统数据库,外部网络(WWW)一般根本无法访问得到,DB'是供外部网络用户访问的相对公开的数据库。同样,DBMS 和 DBMS' 在逻辑上也是不同的数据库管理系统,而且前者的用户是 DB 的所有者和企业内部网络中的用户,后者的用户是 DB'的管理者和外部网络的用户。当然,在物理上,DBMS 和 DBMS',DB 和 DB'并不一定是分开的,这取决于具体的实现方法。

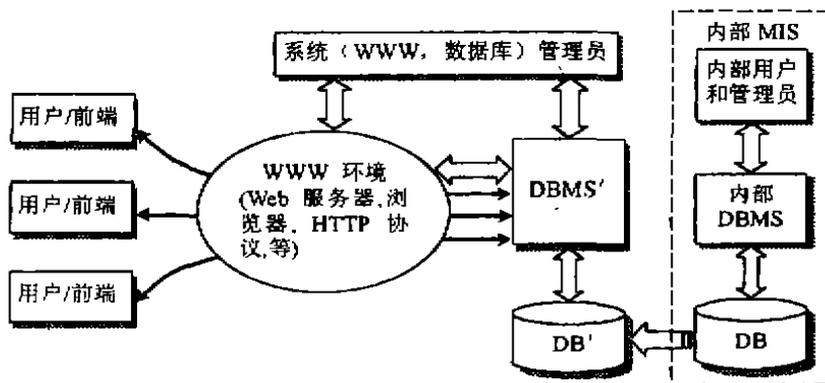


图2 基于 Web 的数据库发布结构框架

图2中 DB 到 DB' 的转换就是发布定义中的  $\xi$  的实现,而用户使用特定或者通用前端通过 Web 访问 DB' 的过程对应于定义中的  $\delta$ 。本文在下面结合具体实例对这两部分的实现进行探讨。

## 3 数据库发布系统的分析和初步实现

### 3.1 $\xi$ 的实现

$\xi$  的实现有两种基本方式:第一种方式下没有专门用来管理被发布的数据库 DB' 的 DBMS',而是借用企业内部网络中的 DBMS,如图3所示。这时候 DB 转变到 DB' 的工作极为简单,只需对要发布的数据库的授权或者存放位置作些调整即可。这种方式的特点是:1)需要有一个通用的数据库驱动管理模块(例如 ODBC/JDBC)来提供对多种类型数据库的访问能力。

这一点带来的好处是可以比较容易地支持多种不同的数据库,只要由数据库厂商提供相应的驱动程序就可以了,缺点是访问数据库的效率会受到影响。2)DB 和 DB' 一般情况下位于同一台计算机上,甚至就是相同的数据库用不同用户的权限进行区分。对于数据库发布者来说这样当然比较简单,而且发布数据库 DB' 的更新也比较方便和及时,但是由于允许外部网络用户访问企业内部网络的计算机和信息,可能带来两方面的缺点:一是安全方面的漏洞,二是不能保证这些内部使用的计算机始终保持开机,从而影响外部网络用户的访问。

另外一种实现方式采用专用于发布的数据库管理系统 DBMS',被发布的数据库直接在 DBMS' 的管理下,如图4所示。相对于第一种方式,这种方案的优点

是:1)Web 服务器中的数据库接口模块只要支持一种或者少数几种 DBMS' 就可以了,由于采用专用的数据库驱动,因此可以获得较高的执行效率,2)数据库发布者不需要保持自己的 DBMS 运行,也不必担心私有数据泄露(DB 是保存在企业内部网络中的)。这种方式下,数据库发布者需要借助实现  $\xi$  的工具软件将要发布的数据库送到 DB',当 DB' 和 DB 是不同的数据库时, $\xi$  还必须负责数据库的转换。当然,转换对发布者来说是完全透明的,另外,这种方式在获得了更高的执行效率和负载能力的同时,付出的代价是发布数据库的更新相对较慢(每次更新必须重新提交),而且必须实现其他数据库到 DBMS' 所支持的数据库的转换,这对于数据库发布系统本身是比较大的工作量。

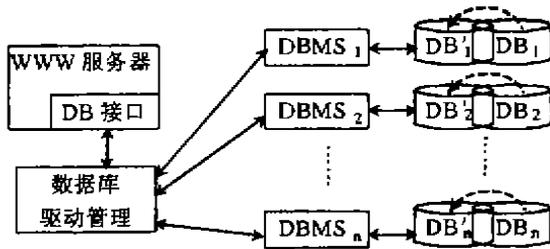


图3  $\xi$  的基本实现方法一

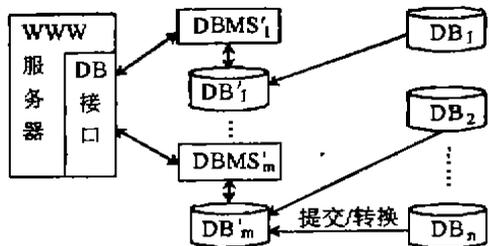


图4  $\xi$  的基本实现方法二

我们的实现采用了第二种方法:发布数据库的管理系统 DBMS' 采用 mSQL,维护着被发布的数据库。数据库发布者可以通过浏览器访问数据库发布站点获得数据库发布接口,通过身份认证后,将本地的数据库提交到发布服务器(运行 mSQL 的专用服务器)上,如果是异种数据库(当前仅限于 DBF 库),发布服务器自动调用相应的转换程序将之转换成 mSQL 格式的数据库。

### 3.2 $\delta$ 的实现

不少文章讨论过  $\delta$  的实现,即如何通过 WWW 访问数据库中的信息。本文在这里通过客户/服务器(C/S)模型的分析讨论不同的实现模式,C/S 模型被认为是商务数据处理的一个时代性的标志<sup>[1]</sup>,尽管今天人们提出了包括三层模型在内的新的模型,这些分析依然具有重要的参考价值。

通过网络访问远程数据库这样的应用可以从功能上分成这样几个组成部分:用户界面程序(连接数据源,形成 SQL 语句,结果数据的处理,中断与 DBMS 的连接等),数据库 API(打开与 DBMS 的连接,读/写数据协议,关闭与 DBMS 的连接等),通信软件,DBMS,数据库等。作为分布式应用系统,这些功能的实现可以分布在客户机或者服务器方实现,根据不同的分配,实际上有三种不同的模型:低能客户/智能服务器模型、智能客户/低能服务器模型和智能客户/智能服务器模型<sup>[4]</sup>。

低能客户/智能服务器模型也就是哑终端/服务器模型,其应用逻辑程序、数据库 API、DBMS、数据库等在主机(服务器)上,终端用户接口和网络/通信软件在客户端。这种模型下客户仅仅是一种显示终端,不能理解和处理所显示的数据,由主机执行用户输入的命令和请求,将处理后的结果送给客户。这种模型在较大型的数据库系统中得到广泛的应用,但是主机一般都要求性能较高的小型机、大型机等。这种模式的好处主要有:1)简单方便。集中存放在主机中的应用程序便于更新和维护。2)安全。用户对数据库所进行的操作都可以由主机进行控制。但是这种模型的局限性也很大:1)服务器压力大。服务器承担的工作太多从而容易成为系统瓶颈。2)应用程序类型受到严重局限。客户端的图形用户界面的能力无法利用,也不支持不同应用程序之间的数据共享。

智能客户/低能服务器模型也就是局域网中常见的 PC 机/文件服务器模型,其中客户端含有用户接口、应用逻辑程序、文件 I/O、网络重定向器,服务器端则只有提供文件访问服务的网络操作系统和数据库文件。这是一种桌面数据库的模式。这种模型下所有的智能集中在客户 PC 机上,而文件服务器比一张可共享的磁盘强不了多少。相比之下这种模型有两个优点:1)可以利用 PC 机的图形用户界面;2)服务器不会成为系统瓶颈。但是缺点也很明显:1)低级的锁定机制;2)最终用户可以很容易地打开文件服务器中的数据文件,并对它的信息部分读或者写。由于文件服务器不具备智能,客户的改动可能使用了不知道文件格式的其他应用程序,从而使文件结构被破坏;3)这种模式没有类似于小型机和主机上的 DBMS 所提供的事务处理和前滚恢复概念;4)最重要的缺陷是造成客户和文件服务器之间大量数据的跨网络传输。客户每进行一次局部查询,文件服务器都要把整个文件而不仅仅是用户所需要的数据传回给用户。

智能客户/智能服务器模型是真正的 C/S 模型。这种模型是对上面两种模型的折衷,旨在将前文中所提到的那些功能合理地分开,将它们放到客户和服务

器端,以达到最好的效果:客户和服务器的资源最大限度地得到了利用;服务器具备智能,因此能够进行集中控制和使用高级术语;同时应用程序逻辑驻留在客户端而不是主机上,这样可以有效地利用客户端所能提供的各种资源而不致造成服务器上的瓶颈效应。

对基于 Web 的数据库发布的  $\delta$  来说,Form+CGI 的形式相当于哑终端-服务器型,服务器端不仅要负责实际计算,还需要把查询结果整理成主页的形式送出,而客户端的显示效果和操作方式完全受限于 HTML 对 Form 和 Table 的支持;用 Java Applet 实现的形式有多种不同的情况,一般来说可以做到更灵活的显示和操作界面,但是由于目前客户端受到 Java 能力的限制,并不能充分利用其图形图像能力,甚至有些应用逻辑也无法在客户端实现,另外由于网络带宽的限制,Applet 不适合于编写复杂逻辑的代码;相比之下,Plug-in/Helper 的实现方式最适合于在 Web 上实现 C/S 模型或者三层模型;针对不同平台的客户端编写的本地代码有很高的执行效率和发挥余地,使得整个系统功能的分布比较灵活。

在我们的实现中,为了提高后端的执行效率,采用在 Web 服务器中嵌入专用 API 模块的方式和 mSQL 数据库引擎交互,前端采用 Plug-in/Helper 形式的专用数据库浏览模块,使得查询界面和结果显示不受浏览器和 HTML 的限制,同时服务器方可以直接把查询结果的数据流返回客户方从而解脱了格式转换的负担。客户端数据库浏览模块的激活依靠自定义的 MIME 类型 application/x-sql,对应的文件扩展名为 db,服务器端,在 NCSA httpd 1.4.2 以前的版本中需要对 HTTP 协议作 DATABASE 方法扩充,但是在 Apache httpd 中由于工作机制的不同(先判断 MIME 类型和模块化),因此实际上已经不需要改变 HTTP 协议了。在这个系统中,被发布出来的数据库只要由一个以 .db 结尾的 URL 标注出来就可以了,当用户通过浏览器点击这样的超链的时候,服务器方通过 MIME 类型识别这样的数据库请求,转交数据库 API 模块,该模块和数据库引擎交互,返回 MIME 类型为 application/x-sql 的文件,浏览器识别该文件的 MIME 类型,激活与该 MIME 类型相对应的数据库浏览前端,用户便可利用该前端进行数据库查询操作。文[5]和[6]介绍了详细的实现细节。

**结论** 并不是通过 Web 访问数据库中的数据就应该称为基于 Web 的数据库发布。对于那些在 Web

上开发的应用和服务导致的数据库应用中,是否采用数据库、采用哪种数据库是由这些应用和服务的要求决定的,开发者关心的是如何让系统和用户用上数据库中的数据,从本文给出的定义和说明中可以看出数据库发布的真实含义,实际上它是区分两者的一个明显的标志之一。

但是不管哪种情况,其结果是通过 Web 进行的数据库访问行为, $\delta$  的实现原则是通用的,系统实现的原则首先是服务效率,其次是用户利益,最终才是开发者和程序员利益。Web 应用的一个特点是用户数目和访问时间的不确定性,任何时间都可能存在大量用户的大量并发请求,因此 Form/CGI 方式一般不是一个很好的选择,除非是限制用户的小规模应用场合,或者服务器本身具有非常高的性能并且服务数据量较小,这时候 Form/CGI 方式对于开发者和维护者来说是最容易的;Applet 的使用非常好地照顾了开发者的利益:只需要一次编程,而且系统发行和维护非常方便等,但是采用 Applet 的时候应该考虑对最终用户利益的损害:即必须为重复下载这些程序而付出时间、费用的增加和执行效率的降低。因此实际上 Applet 只适合于小巧简单的应用场合,而不适合于比较复杂的应用逻辑;Plug-in/Helper 和上两种方式具有互补的优势。采用这种方式当然给开发者带来了很大的工作量,但却获得了最佳的系统性能,基于这一点考虑,本文的具体实现采用了这种方式。

当然,这些模式并不是互斥的,因为基于 Web 的应用方式不仅给用户也给开发者提供了非常灵活的选择。

#### 参考文献

- 1 唐常杰,等.论数据库技术与 Web 技术的融合.计算机世界.专题,1998.3:D6~D7
- 2 刘欣然.基于 Web 的计算资源发布研究:[博士论文].哈尔滨工业大学,1998
- 3 Press L. The Next Generation of Business Data Processing. CACM,1999,42(2):13~16
- 4 Geiger K. Inside ODBC. Washington: Microsoft Press,1995
- 5 薛慧.基于 Web 的数据库发布前端及相关技术的研究与实践:[硕士论文].哈尔滨工业大学,1998
- 6 金舒元. WEB SERVER 嵌入技术及相关问题的探讨与实践:[硕士论文].哈尔滨工业大学,1998