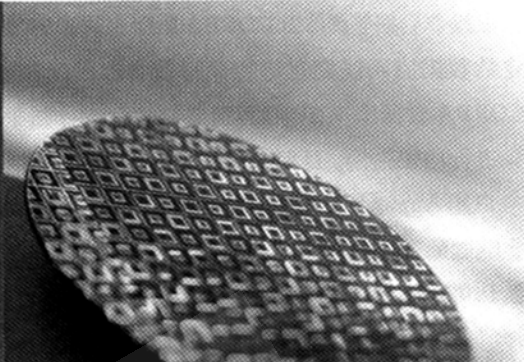


基于CORBAWEB的 异构数据库联合使用



刘健 梁允荣 (北京理工大学计算中心 100081)

摘要: 未来对各种异构数据库系统的使用将包括 WWW 环境和 CORBA 标准。结合 WEB 和 CORBA 技术, 可以给用户提供具有友好界面的、面向对象的和高效率访问异构数据源的应用。本文讨论了数据库技术、WEB 技术和 CORBA 技术的集成, 并进一步阐述了如何使用 CORBA 技术访问网络上的异种数据库和这种技术的优势以及应用前景展望。

关键词: CORBA CGI IIOP CORBAWEB DATABASE

1 WWW 数据和 CORBA

国际互连网是一个全球范围的, 访问各种分布式数据源的系统。各种数据源包括一些超媒体文档(如各种 HTML 页面), 各种异种数据库, 和许多驻藏 WEB 服务器上的在可执行程序(如搜索引擎和电子商务服务程序)。

当今, 从 WEB 服务器访问外部的各种数据库资源, 主要通过通用网关程序(CGI)来实现。这种方式主要的弊端是, 在 WEB 服务器上, 每当有一个对数据库的访问请求到达时, 一个单独的 CGI 进程就被启动, 这样随着访问量的不断增大, 服务器上的 CGI 进程数目也不断增大, 严重影响了服务器的效率, 使其处理速度随着请求数的增加而锐减。另外, HTTP 是一种无状态(stateless)的协议。每次下载一个页面, 客户机就与服务器完全脱离, 服务器无从得知客户机的信息, 更无法知道脱离后的客户机正在进行的动作。其他一些 CGI 的扩展, 如 cookie、Microsoft Internet Services API (ISAPI)、Active Server Page (ASP) 和 Netscape Server API (NSAPI) 也不能从根本上解决这个问题。

随着 Internet、Intranet 和 Extranet 的发展, 越来越多的数据库系统会和 WEB 系统集成在一起; 在包括电子商务的各种 WEB 数据库应用中, 各种对于数据库的请求必须高效完整的得到处理。处理的安全性和各种商业数据的完整性等也必须得到保证。

通过把一个统一的 WEB 数据库访问界面和轻便的可移动客户端有机的集成到 WWW 中, 就可以达到这个目标。这样可以在同一个系统结构中提供一系列的对象来分

别访问各种 WWW 数据库, 这些对象一旦和用户客户端连接, 就能保持这种连接, 直到交易完成。新的访问数据库的对象能够被无缝的插入到系统中而原系统不做任何改动。对象管理集团(Object Management Group, OMG)的公共对象请求代理模型(Common Object Request Broker Architecture, CORBA)提供了一个很好的模式来封装和构建各种 WEB 应用。它能提供访问 WEB 数据库时所需要的各种特性: 面向对象性、模块性、灵活性和分布性等。

2 WEB 中基于 CORBA 的异构数据库应用

基于 CORBA 的 WEB 异构数据库应用系统意在为最终用户提供一个统一的友好的数据库应用接口, 并屏蔽掉各种网络中的异构成分, 包括数据库管理系统的异构、网络操作系统的异构、硬件平台的异构以及网络拓扑结构的异构等。这使得在网络上浏览和修改各种异构数据库象用户浏览 HTML 页面一样容易和方便。

任何形式的浏览器都可以作为系统的客户端, 系统通过在浏览器上运行的 Applet 和在服务器上运行的 CORBA 数据库服务对象的交互, 完成各项功能。浏览器负责完成和用户的交互, 并把用户的需求以固定的格式传到服务器; 在 WEB 服务器上驻留的 IIOP 防火墙接收到用户的 IIOP 请求消息, 进行一定的过滤和控制后, 把请求发给固定的 CORBA 数据库 WEB 服务对象, 建立起一个客户和服务器的常连接; CORBA 数据库 WEB 服务对象根据系统的全局数据字典, 对客户请求进行任务分解, 并调用相应的 CORBA 数据库服务对象进行具体的数

数据库操作；各个具体的CORBA数据库服务对象负责和特定的数据库系统交互，完成对最终数据库数据的访问；CORBA数据库WEB服务对象接收到操作结果，并将结果进行格式转换，返回给客户，完成一次数据库应用。当客户端Applet完成所有操作时，客户和服务器的常连接也随之断开。

2.1 系统结构

如图 1 所示, WEB 上基于 CORBA 的异构数据库联合使用结构主要包括三部分: 前台用户接口、中间全局逻辑控制层和局部数据库访问层。前台用户接口(Applet)主要用于和用户的交互; 中间全局逻辑控制层(CORBA DB WEB Server、查询分析和查询翻译)主要用来屏蔽掉各种异构成分, 提供一个统一的逻辑数据库视图; 局部数据库访问层(局部数据对象)完成最终的数据库数据访问。

(1) WWW客户端: WEB客户端从服务器上下载带有CORBA数据库客户对象的Applet, 在对象中有访问远程数据库服务对象的代码。Applet使用CORBA IIOP协议来得到远程CORBA数据库服务对象的服务。

(2) IIOP防火墙: IIOP防火墙安装在WEB服务器上,它能够过滤、控制和记录所有的IIOP信息,并可以把CORBA服务请求,通过WEB服务器,转接到提供特定服务的CORBA对象上,即使这些对象没有驻留在WEB服务器上。在CORBA数据库WEB环境中,IIOP防火墙把请求转接给CORBA数据库WEB服务对象。

(3) CORBA 数据库 WEB 服务对象：从客户端 Applet 到达的数据库服务请求，经过 IIOF 防火墙的过滤和转接后，即到达 CORBA 数据库 WEB 服务对象。服务对象根据系统全局数据字典，分析用户的请求，把用户请求转换成全局 SQL (GSQL) 查询语言，并把转换完成的查询语言递给查询分析对象。CORBA 数据库 WEB 服务器同时也接收所有的局部 CORBA 数据库服务对象返回的查询结果，并把结果综合和装配，返回给用户。这样，CORBA 数据库 WEB 服务器就在 WEB 和 CORBA 数据库环境之间充当桥接器的作用。系统安全控制也在这个对象中给予实现。

① 全局数据字典：系统全局数据字典为系统查询分析和查询翻译提供统一的全局数据视图。数据字典包括全局库的所有信息和局部库的部分信息：全局库信息包括所有的全局数据库的属性和所有全局数据表的详细描述；局部库信息包括每个局部库的名称、数据库类型、数据库所在的主机的网络地址、以及权限控制信息等。查询分析和

查询翻译对象根据全局数据字典来进行具体计算处理。

② 查询分析：查询分析对象对接收到的全局查询语句(GSQL)进行语法和词法分析，如发现错误则把结果回传给数据库WEB服务对象，否则则把分析完成后的结果传给查询翻译对象进行翻译。

③ 查询翻译：首先把分析对象发出的全局数据查询分解成针对相应局部数据库系统的子查询，因为各个局部数据库系统的查询语言与全局查询语言可能有较大的不同，因此分解出的子查询必须再翻译为各局部数据库系统的查询语言。

(4) CORBA 数据库对象访问层：从系统全局层访问局部数据库对象，可以使用多种不同的方法：①直接绑定到服务对象。这种方法是一种最简单的访问 CORBA 对象的方法，对于一个特定的客户服务请求，绑定某些特定的 CORBA 服务对象。②使用 CORBA 动态请求界面(DII)。CORBA DII 允许应用程序发布对任何对象的服务请求，即使应用程序编译的时候并不知道该对象的服务界面。③使用 CORBA 命名服务(NS)。CORBA NS 是一种最简单但却是最有用的 CORBA 服务(CORBA Services)，它通过把对象和一个具体名称绑定在一起，允许用户使用命名服务通过解析名称而得到对象的引用。CORBA 命名服务同时可以用来实现对象的负载均衡，通过负载均衡，能够更有效的利用系统内的各种资源。

(5) 标准 CORBA 环境：标准 CORBA 环境包括所有的 CORBA 规范组件，包括各种 CORBA 服务、CORBA 工具(facilities)和界面仓库(Interface Repository)、实现仓库(Implement Repository)等。

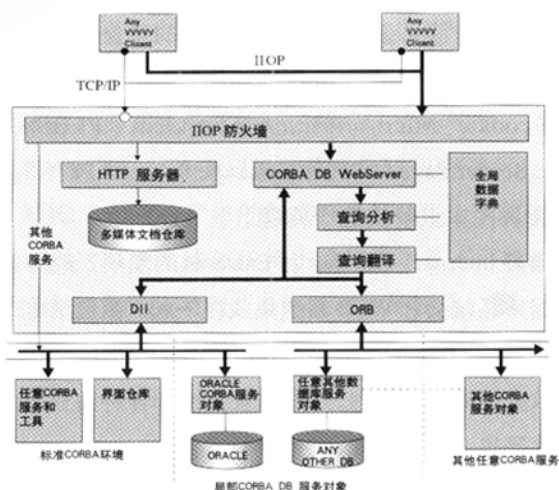


图 1 CORBA DB 结构图

(6) 局部 CORBA 数据库服务对象: 局部数据库服务对象负责完成最终的数据操作, 并把结果返回给 CORBA 数据库 WEB 服务对象。可为每一个不同种类的数据库设计服务对象并放置在物理数据库实际存在的机器上, 将局部数据库服务对象同联合使用的其他服务对象相分离。根据访问数据库系统的不同划分成多个对象, 可以充分利用各数据库厂商自行提供的访问接口(大多为 C 接口), 能大大提高查询的速度。为了维护局部数据库设计和扩充的独立性, 使得系统各个数据库系统更为灵活, 一部分系统数据字典放在局部数据服务层, 内容主要包括从全局表到局部表的映射, 以及相关全局表与本地表的所有对应关系, 如名称、表结构等。因此, 一部分 SQL 语句的翻译也必须在本地进行。另外, 数据库服务对象还要提供查询优化处理的功能。

2.2 WEB 数据库应用中 CORBA 技术的优势

使用 CORBA 技术架构和访问 WEB 数据库有许多优点:

使得对 WEB 分布式数据库的访问简单高效: 在这种分布式应用环境下, 可以使用 WEB 浏览器访问多种异构数据库。这种访问仅需要用户有一台内嵌 JAVA 虚拟机的入网终端。在网络上, 所有的操作都是在 CORBA 规范下进行, 因为 CORBA ORB 隐藏了网络环境下的各种异构性(包括网络操作系统、硬件平台、网络拓扑结构的异构等), 可以很容易的把系统内的各种组件和对象分布在网络中不同的平台之上。同样由于在服务器端使用了 CORBA, 系统各个服务对象的移植也变得非常简单。通过把 CORBA 服务对象分布在不同的主机和 CPU 上运行, 能够更好的提高系统的交行性, 从而提高系统的总体效率。

使用 CORBA OTS 服务实现分布式事务的处理: 在 WEB 上对多个异构数据库的访问是一个分布式事务处理过程, 需要一个事务管理器来完成事务处理, 以使得对全局数据库的处理可以跨越多个应用程序、处理进程和系统主机。对象管理集团(OMG)的对象传输服务(OTS), 是建立在 X/Open 公司“分布式事务处理”(DTP)基础上的 CORBA 事务管理的标准。通过 CORBA, 能够提供和 DTP 参考模型类似的分布式事务处理, 提供一种面向对象的异构数据库事务管理, 以实现数据的一致性和完整性。

主动服务: 通过在客户端和服务端使用 Callback 和 Oneway 方法, 在客户端嵌入的 CORBA 对象能够在服务器上注册, 对象的方法也可以被服务器对象调用。这样就

建立了一个对等的(peer to peer) CORBA 应用程序。用户浏览器和后台 CORBA 服务器互为客户端和服务端, 没有明显的客户端、服务器分别。这样不仅提高了效率, 而且使 WEB 数据库服务器由被动服务转为主动服务。

2.3 基于 CORBA 应用的开发工具和运行环境

开发基于 CORBA 的应用需要使用第三方提供的开发支持工具, 目前主要有 IONA 公司的 Orbix 系列和 Visigenic 公司的 VisiBroker 系列产品。同时还需要传统的编程工具, 如 Microsoft 的 Visual C++ 和 Inprise 公司的 Delphi 和 C++ Builder 等, Orbix 和 VisiBroker 等 CORBA 支持产品提供建立 CORBA 应用程序所需的基本库和对 CORBA 中各种组件的实现, 而传统的开发工具则负责应用程序最终的编译和连接。目前 IONA 和 Visigenic 公司分别提供了基于不同语言的 CORBA 实现产品, 如 OrbixWeb 就是 IONA 公司实现的从 IDL 到 Java 语言的映射, 而 Orbix 则是 IDL 到 C++ 语言的实现, 而在不同的操作系统上也有不同的中间工具。同时一些 CORBA 应用辅助产品, 如 WonderWall 等, 进一步完善了 CORBA 的使用和开发。

当 CORBA 应用投入使用后, 相应的 CORBA 服务程序, 如 Orbix Daemon 等要在 CORBA 服务器端运行, 以提供程序正常运行所必需的 CORBA 底层服务。另外如果客户端有一些特殊的运行要求, 如要使用 CORBA 的 Locator 服务, 则在客户端也必须要有相应的 CORBA 底层服务进程。

2.4 CORBA 应用前景

虽然目前 CORBA 企业级的应用比较多, 而且也比较成熟, 但由于 CORBA 程序的开发有一定的难度, 再加上开发工具的相对欠缺, 所以大规模的基于 CORBA 的开发还没有形成。当前已经出现了针对 CORBAWEB 开发的 Script 语言 CORBAScript, 这进一步简化了 CORBA 应用程序开发的难度和时间, 另外一些 CORBA 产品实现厂商也在不断的提供更简单, 更容易使用的开发工具。总之, 随着 CORBA 的各种使用工具和其他分布式技术的发展, 基于 CORBA 的应用必将越来越普及。■

参考文献

- 1 The Common Object Request Broker Architecture and Specification, Revision 2.2, by OMG & X/Open, 1998.2. <http://www.omg.org>
- 2 Orbix Programmer's Guide IONA Technologies PLC 1997.10
- 3 CorbaWeb: A WWW and Corba Worlds Integration, Philippe Merle, Christophe Gransart, Jean-Marc Geib.