

文章编号:1006-7329(2000)03-0049-05

# 基于 CORBA 的网络数据库技术\*

TP3/P2

49-53

喻 玲, 齐晓红

(重庆建筑大学 网络中心, 重庆 400045)

**摘要:**首先对比通过 Web 访问数据库的几种技术,引出一种新的技术:CORBA;简述了 CORBA 的几个重要的特点,在此基础上以人事管理系统为例分析了 CORBA 的工作机理,提出了一种基于 CORBA 的网络数据库的成功实例。网络数据库。

**关键词:**CORBA; ORB; IDL; 对象; Web 服务器; 分布式数据库系统

**中图分类号:**TU392

**文献标识码:**A

随着网络技术和数据库技术的飞速发展,人们通过 WWW 获取信息越来越普遍。但这种以 HTML 文件为基础的信息组织机制过于庞大,相对杂乱,不便于管理、更新和查询,需要借助于发展较为成熟的数据库技术加以改善;与之相应的,用传统的 DBMS 建造的数据库资源正急需通过网络来发布和共享,使之充分利用。因此数据库技术和网络技术(尤其指 Web 技术)的融会成为当前研究的热点。

## 1 对比

人们为了解决上述问题,提出了基于 WEB 的分布式数据库系统的体系结构:B/S 的三层分布结构,见图 1:

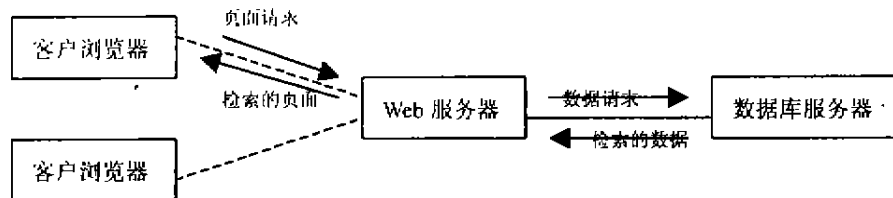


图 1 浏览器/服务器三层结构

从这种结构我们会发现,最关键的问题是如何通过 Web 访问数据库中的信息,即建立 Web 客户与数据库服务器之间的通信接口。近几年来,人们已设计了几种方法和途径,大致分为:CGI 技术、NSAPI 与 ISAPI 结合的 ODBC 技术、Java JDBC 技术、ASP 结合 ADO 技术。

CGI 作为最早的数据库与 Web 的连接方式,得到了广泛的应用,但采用 CGI 技术执行效率低,为了克服 CGI 的局限,一些 Web 服务器厂商开发出专用的 API(Application Program Interface),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 技术采用的是一种线性编程方式,因此效率比 CGI 高,而且可以使用 Web 服务器本身的安全机制,但它必须依赖于特定的服务器。Java 是一种面向对象的程序语言,尤其用于不同种计算机网络上的设计。由 SUN 公司提出的 JDBC 技术实现了数据库与 Web 的连接,同时也充分利用了 Java 语言的优点,因此有较好的健壮性和灵活性。ASP 是微软的 IIS3.0

\* 收稿日期:2000-04-27

作者简介:喻玲(1972-),女,重庆人,硕士生,主要从事网络和网络数据库研究。

的一个部件,它采用 VBscripts 或 Java scripts 脚本语言开发,将 HTML 和脚本开发融合在一起,支持面向对象编程,支持 ActiveX 控制,因 ASP 文件中的脚本在服务器端执行,效率高,灵活性好。尽管上述技术各具特点和优点,但基本上都需要设计者有一定的编程基础;此外,对于多个异构的分布式数据库系统的访问来说,功能就相对弱多了;同时设计出来的模块可移植性、可重用性都较差。如何寻求一种既简化编程,又支持客户访问网络数据库系统的方法呢? CORBA 规范的出现以及 CORBA 技术的应用无疑给人们提供了一条解决问题的新的有效途径。

## 2 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互操作,目的是使基于对象的软件部件在分布异构的环境下可重用、可移植和可互操作。

CORBA 规范是针对 OMA(对象管理体系)参考模型中的对象请求代理 ORB 而制定的,OMG OMA 由如图 2 所示的四个构件组成。

1) 对象请求代理 ORB:使对象在分布式环境中透明地收发请求和响应,它是构建分布式对象应用,使应用在不同层次的异构环境下互操作的基础。

2) 对象服务:是为使用和实现对象而提供的基本服务集合。如为创建对象、对象访问控制、对象跟踪和对象引用等提供一套标准函数。

3) 应用程序对象:为最终用户提供一套完成特定任务的对象。

4) 公共工具:为许多不同的应用程序提供一套一般用途的应用程序功能,例如数据库存取工具,文件打印工具。

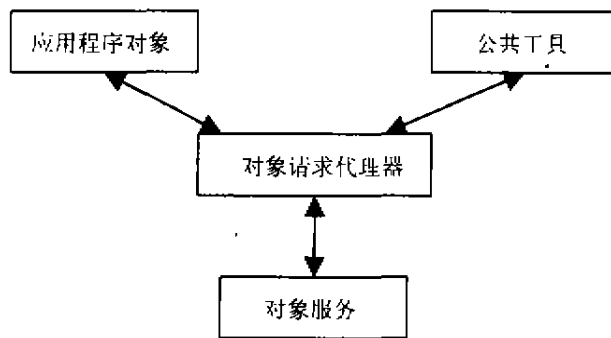


图2 对象管理体系的4个部分

## 3 CORBA 的技术特点

CORBA 规范定义了 IDL 语言及映射、单个 ORB 和 ORB 之间互操作机制。它具有这样一些技术特点和优点:

1) 引入了代理(Broker)的概念。一个代理至少有三个方面的作用:完成对客户方提出的抽象服务请求的映射;自动发现和寻找服务器;自动设定路由,实现到服务器方的执行。这样用户在编制客户程序时就可以避免了解过多的细节,而只要完整地定义和说明客户需要完成的任务和目标。

2) 实现了客户方程序与服务方程序的完全分离,这与面向过程调用机制为基础的客户/服务器模式根本不同。客户将不再同服务器发生直接联系,而仅仅需要与代理进行交互。客户和服务器之间就可以有更加灵活的关系存在。

3) 将分布式计算同面向对象的概念相结合,可以提高软件重用率、控制冗余度等,带来诸多好处。

4) 提供了软件总线的机制,所谓软件总线是指 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,而这个接口规范独立于任何实现语言和环境。

5)分层的设计原则和实现方式。CORBA 规范仅定义了 ORB 中需要用到的基本对象类,封装了相应的数据结构和方法,而面向应用的对象定义则可以在 OMA 的应用对象或应用开发环境中逐步分层定义和实现。

### 4 基于 CORBA 的事例分析

CORBA 的系统体系结构如图 3, 其中的构件或接口的功能在此就不一一详述,我仅以某公司存在的分布式人事管理系统为例简述基于 CORBA 的数据库系统的实现机理。

1) 根据要解决的人事管理系统问题建立抽象模型,在此基础上建立对象模型,然后用 OMG IDL (接口定义语言)定义接口、变量、数据类型等。

//人事管理系统中的 OMG IDL 部分 corba1.idl

```

Module corba1
{
  //定义变量
  typedef long idnum
  typedef long deptnum
  .....

  //数据类型定义
  struct employeeinfo
  {
    long id
    string name
    string phone
    .....
  }

  //定义接口
  interface employee
  {
    { attribute employeeinfo employee _ data;
    .....
    Void promote (in char new _ job _ class);
    Void dismiss (in dismissalcode reason,
                  In string description);
    Void.....
  }
  .....
}

```

并将接口定义放在接口仓库中以备后用。

例如 >idl \_load corba1.idl

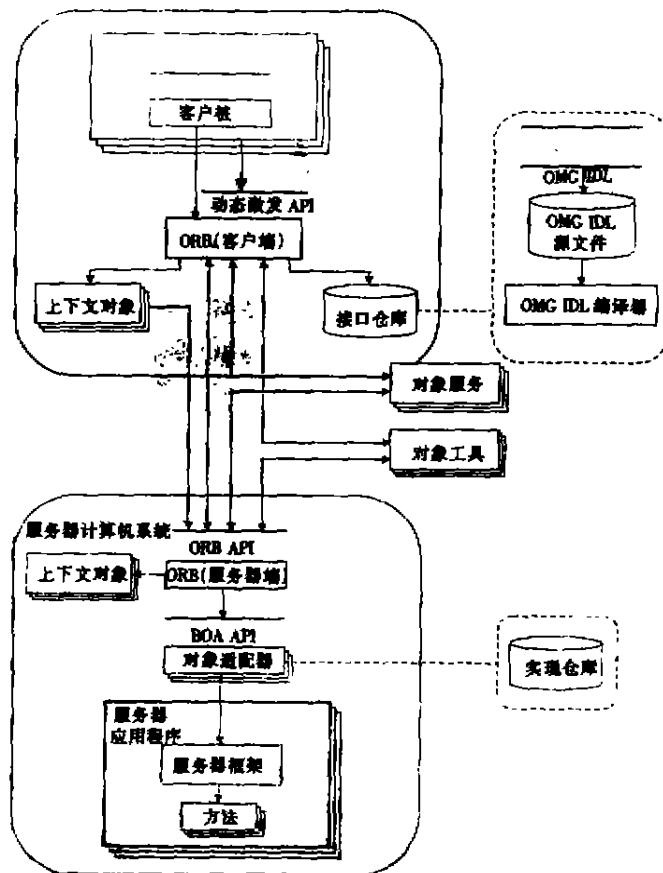


图3 CORBA 系统体系结构

2) 用 OMG IDL 编译器将其生成客户桩、头文件以及服务器框架。

```
例如: >idl_compile corbal.idl _c corbal_client.c
      >idl_compile corbal.idl _s corbal_socket.c
```

3) 客户端发出请求

```
例如: void corbal_employee_promotee
      { corba_object employee1
      corba_enviroment * enviroment
      corba_char new_job_class
      .....
      }
```

请求所传递的信息类似于结构化程序调用函数时传递的实参。

客户机应用程序采用桩类型激发或动态激发向服务器发送请求,在此采用的是桩类型激发。

4) 在客户端的 ORB 处理请求的激发。

当应用程序为某个对象上的某个操作向 ORB 发送请求时,ORB 检验参数是否跟接口一致,并把请求转发给服务器。

5) 在服务器端,ORB 接受方法调度请求,激发服务器框架中的方法调度器,引导输出参数,并完成激发。

在图 3 中,可以看到客户机端的 ORB 和服务器的 ORB,实际上只是同一个 ORB 代码驻留在客户机和服务器。在高版本的 CORBA 中,一个 CORBA 系统可能会有多个 ORBs,多个 ORBs 可以含有对象引用的不同表示以及不同的激发风格。

6) BOA 基本对象适配器是最常用到的对象适配器。对象适配器处理与 ORB 相关的普通任务,如提供对象引用、激发实现等。

例如://创建对象

```
CORBA_BOA_create
//激活对象,准备就绪
CORBA_BOA_obj_is_ready
```

7) 服务器框架是一个与语言相关的映射,提供调度请求到适当方法的必要代码。服务器应用程序就是利用此框架文件将客户机操作映射到服务器实现的方法上。例如对象 employee 定义的操作 promote 和 dismiss 对应服务器端方法 emp\_promote 和 emp\_dismiss。当服务器执行 CORBA\_BOA\_create 时,接口就和实现相关联。

在此必须说明,接口与实现并非必须一一对应,我们可以根据需要构建一个接口对应多个实现中的一个,或者一个接口对应多个实现。

## 5 总结

以上我们对基于 CORBA 的数据库系统的工作机理仅作了简单的介绍,如果要实际规划一个基于 CORBA 的分布式数据库系统时,设计就更加复杂,需要考虑多方因素,比如分布异构资源的透明性、并发控制、全局查询如何转化为局部查询等等。对于这些问题国内外已经有了一定的解决方案。

目前已成功地将 CORBA 应用于 WEB 数据库系统中,实现了 WEB 客户与数据库服务器之间的访问。参见图 4,各个部分具体的实现在此就不详述了。

总之,CORBA 技术以其不容置疑的优势正在不断发展,基于 CORBA 的分布式数据库系统技术有着广泛的应用前景,有待进一步深入研究和开发。

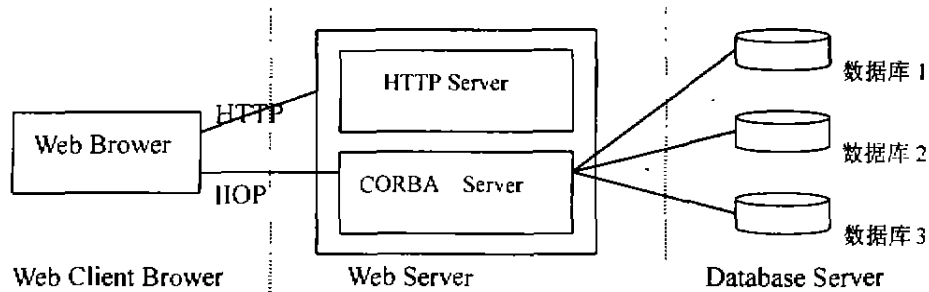


图4 三层分布式B/S体系结构

### 参考文献:

- [1] M. Rog 著, CORBA 教程[M], 李师贤等译, 北京:清华大学出版社, 1999
- [2] 汪芸, 顾冠群, CORBA 技术综述[J], 计算机科学, 1999; 26(6): 1~6
- [3] 薛慧, 代亚非, 基于 CORBA 的 Web 数据库发布技术[J], 计算机科学, 1999; 26(6): 67~69
- [4] 刘锦德, 苏森, CORBA 技术的新发展[J], 计算机应用, 1999; 19(5): 1~3

## Networking Database Technology Based on CORBA

YU Ling, QI Xiao-hong

(Network Center, Chongqing Jianzhu University, 400045, China)

**Abstract:** This paper firstly introduces a new technology: CORBA by comparing several technologies of Web accessing database, then a few main features were briefly described. On this basis, the CORBA working function was analyzed. At last, an example of successful CORBA-based network database was presented.

**Keywords:** CORBA; ORB; IDL; object; Web server; distributed database system