

基于 Internet 的构件化管理信息集成平台 ICM IIP 的研究与实现*

刘 惠 贾 焰 窦 文

(国防科技大学计算机系 长沙 410073)

摘 要 论述了基于 Internet 的构件化管理信息集成平台 ICM IIP 的设计思想和实现技术,包括遵循 CORBA 2.0 标准的 Java ORB 的实现,以及如何使用 Java ORB 集成遗留的关系数据库应用系统,并将遗留应用推向 Internet 的方法。

关键词 对象请求代理, Java, Internet, 数据库

分类号 TP392, TP393

The Research and Implementation of ICM IIP Based on Internet

Liu Hui Jia Yan Dou Wen

(Department of Computer, NUDT, Changsha, 410073)

Abstract The paper discusses the research and implementation of ICM IIP based on Internet, including the implementation of Java ORB complying with CORBA 2.0 specification, and the methods of how to use Java ORB to integrate the legacy application developed on relation data base system and lead them to Internet/Intranet.

Key words Object Request Broker, Java, Internet, Database

随着网络软硬件技术的飞速发展,越来越多的企业希望能够利用 Internet 来开展业务,电子商务的发展也急需大量能在 Web 上协同工作、交互访问的应用程序,这些需求对 WWW 技术提出了更高的要求:(1) 提供超越 Client/Server 结构的更好的体系结构;(2) 在 Internet 环境中保证数据的安全性和完整性;(3) 稳固的、高可伸缩的、高可重复利用的应用开发平台。

因此在分布异构环境下,构造一种全新的应用平台,不仅可以把遗留应用系统推上 Internet,而且新开发的支持 WWW 的分布应用系统可以随时添加到系统中,已经成为 Internet 发展的一个热点问题。

从 90 年代初开始,面向对象技术被广泛应用于分布计算领域,其中 CORBA (Common Object Request Broker Architecture) 标准已经成为建立分布式应用框架和软构件的技术标准。本文将讨论以 CORBA 标准为核心的基于分布对象技术的 Internet 构件化信息管理平台 ICM IIP (Internet Component Management Information Integrated Platform) 的解决方案。

1 基于分布对象技术的管理模型

分布对象技术采用面向对象的客户/服务器结构,将分布在网络上的全部资源按照对象的概念进行组织,每个服务对象有定义明晰的访问接口;根据服务对象的访问接口来使用服务器上所提供资源的应用被称为客户,而服务器则创建和管理维护对象实体并负责受理客户对象的服务请求。CORBA 正是一个规范分布对象之间如何定位和通讯的标准。CORBA 标准通过对象请求代理(Object Request Broker)机制,实现了客户对象和服务对象之间的通讯;通过定义 Internet 交互对象代理协议 IOP (Internet Inter-ORB Protocol) 和接口定义语言 IDL (Interface Definition Language),提供了分布对象间的互操作能力。图 1 反映了依据 CORBA 标准,采用面向对象技术的分布式系统的典型结构。其中,对象

* 863 计划软件重点课题及国家自然科学基金重点课题资助。
1999 年 3 月 20 日收稿
第一作者:刘惠,女,1971 年生,讲师

请求代理 ORB 是 CORBA 标准支持客户访问异地服务对象的核心机制; IIOP 协议是 ORB 内部遵循的支持各种 ORB 在 Internet 上进行互操作的协议标准; 对象服务包括各类基本的系统级服务, 例如名字服务和接口池服务等等; 公共设施包括支持分布式系统有效工作和高效开发分布式应用各类面向领域的高层服务和开发工具, 如 GUI 服务、数据库服务等; 应用对象则涉及各种在对象服务和公共设施的协助下完成相应应用逻辑的应用软件。

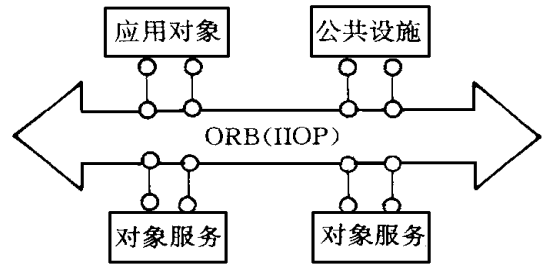


图 1 分布对象管理结构 OMA

Fig. 1 Distributed objects management architecture

这种面向对象的分布式系统结构支持分布式应用开放特性, 从而为实现基于 Internet 的构件化管理信息集成平台 ICMIP 提供了有效的支持: 每一个服务对象由 CORBA 服务器创建和维护; ICMIP 作为客户对象, 利用标准的名字服务和接口池服务, 解决服务对象动态加入系统的问题, 并且根据服务对象所提供的操作接口通过 ORB 管理和使用应用服务对象。

2 ICMIP 的设计原理

如图 2 所示, ICMIP 采用一种基于用户界面层、中间层和应用对象层的三层体系结构实现方案。在 ICMIP 环境中, 应用程序可以划分为客户方程序和服务器方程序。客户方程序采用 Java Applet 语言进行开发, 然后使用 IDL to Java 编译器对其进行编译, 实现与 Java ORB 的绑定, 并将其驻留在 HTTP 服务器中; 服务器方的应用程序采用 Java 或者包括 C++ 语言开发, 建立同本地 ORB 的连接。

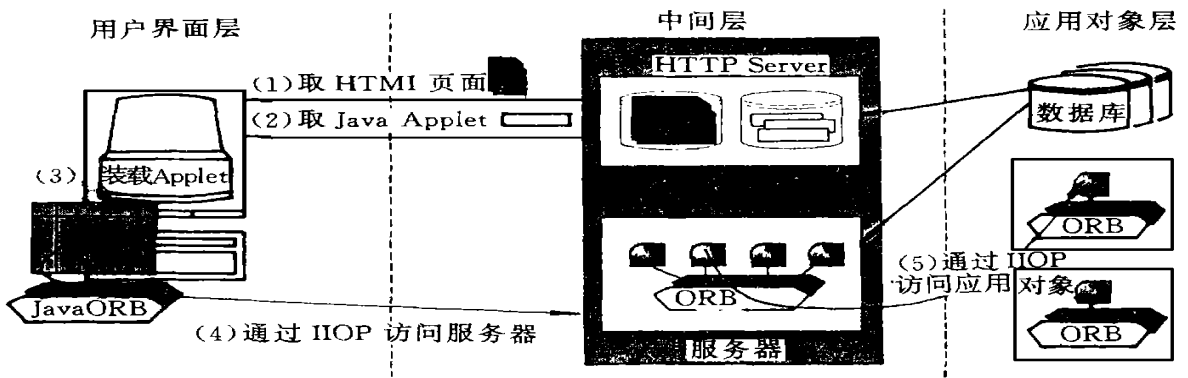


图 2 与 Web 技术相结合的 ICMIP 解决方案

Fig. 2 ICMIP solution combined with Web

ICMIP 的用户界面层采用标准的 WWW 浏览器。中间层包括 HTTP 服务器和管理服务器, 管理服务器运行相关的系统服务对象(例如名字服务、事件服务等), 而 HTTP 服务器负责管理 HTML 主页和访问应用对象的 Java Applet, 以支持用户界面能够按照 WWW 的方式进行工作。应用对象层是用户定义的服务对象以及数据库服务器。根据遵循 IIOP 协议的 Java ORB 和 Java Applet 解释器, 用户可以标准的浏览器方式使用该平台, 其典型工作过程可以描述如下:

- (1) 通过浏览器从 HTTP 服务器上下载管理应用的 Web 页面以及访问管理对象的 Java Applet;
- (2) 浏览器装载 Java Applet;
- (3) 通过 Java Applet 解释器, 遵循 IIOP 协议访问管理服务器上的管理对象;
- (4) 管理服务器上的管理对象访问被管理对象。

ICMIP 实现上的三层结构, 既解决了传统浏览器难以灵活操纵各种管理服务以及被管理对象的局限, 又充分继承和发扬了界面统一、易于添加新的应用等 Internet/ Intranet 的特点。

3 IIOP 协议的实现

GIOP(General Inter-ORB Protocol) 协议定义了不同 ORB 之间进行通讯的标准传输语法(数据的低级表示标准)和消息格式,而不依赖于高层的RPC 机制。该协议定义了从IDL 数据类型映射(Mapping)到可以被任何面向连接的网络传输的消息标准,即通用数据表示(Common Data Representation, 简称 CDR);同时,它还定义了可互操作的对象引用表示(Interoperable Object Reference, 简称 IOR),以便信息可以在不同的支持 CORBA 2.0 标准的 ORB 之间进行交换。而 IIOP(Internet Inter-ORB Protocol) 协议则完成 GIOP 协议到 TCP/IP 协议的映射,是 OMG 组织为支持 CORBA 2.0 标准的 ORB 对象实现之间进行互操作而制订的标准。

在分布式环境下,只有通过所要访问对象的对象引用才能准确定位和使用该对象。在具体应用中,当客户方得到服务方实现对象的对象引用后,首先调用 ORB::string_to_object 方法在客户方产生服务对象的代理对象(Proxy);代理对象负责将客户对象发出的服务请求按照 IIOP 协议格式进行编码,形成请求消息,发送到服务方;服务方接收到此请求消息后,首先对此消息进行解码,然后由对应实现对象实现客户方的请求后,再形成应答消息返回到客户方。下面举例说明 IIOP 消息实现后的格式:

假设一应用的 IDL 接口定义如下:

```
Interface Cubit {
    struct many {
        octet    o;
        long     l;
        short    s;
    };
    many Cube_struct(in many arg);
}; // 该应用实现对一个 many 的构造数据类型的每一元素进行立方操作
客户方调用 Proxy->Cube(m); // 其中, m.o = m.l = m.s = 117;
向服务器方发送的消息为(按 IIOP 协议编码):
```

hex 表示:

```

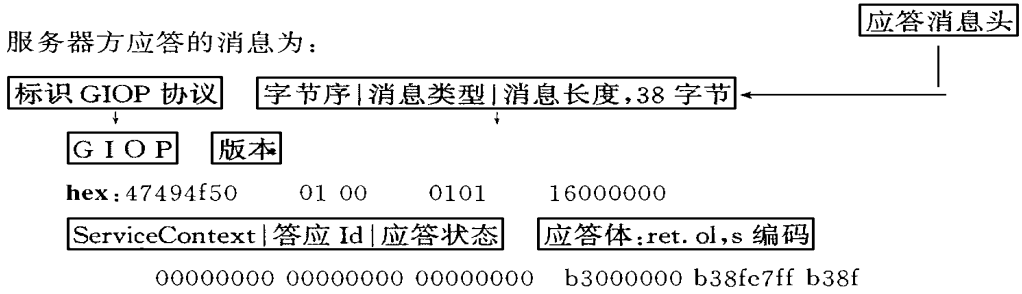
标识 GIOP 协议   字节序|消息类型   消息长度,86 字节
  ↓              ↓              ↓
  GIOP   版本
47494f50 01 00   0100   56000000 // 此为消息头

ServiceContext|请求 Id   是否返回|字节补齐   Object key 编码:
00000000 00000000   01 00e71e   26000000

长度+OB/ID+NUM/IDL: Eng. SUN.COM/Cubit:1.0
442b4e55 4d004944 4c3a456e 672e5355 4e2e434f 4d2f4375 6269743a

+0   操作名编码:长度+cube struct   安全机制
30000000 0c000000 63756265 5f737472 75637400 00000000
//以上为请求头

//以下为请求体,即输入参数的编码
m. |字节补齐   m. l   m. s
8b f96600   8bffffff 8bff
```



由上可见, 按照 GIOP 协议和 IIOP 协议, 形成的消息格式其实就是 GIOP 模板定义中的数据遵循通用数据表示(CDR)进行编码后的结果。因此, 在分布式应用中, 只要 Client 方和 Server 方支持 IIOP 协议格式, 便可以实现不同平台、不同语言的 ORB 产品之间的交互。

4 SQL 服务实现

实现 ICMIP 平台时, 客户方使用 Java ORB 平台, 服务方使用 StarBus 2.02 for C++ 平台。前者为纯 Java 语言的 CORBA 2.0 标准实现, 而后者为 CORBA 2.0 标准的 C++ 实现。数据库服务器使用银河数据库系统 GalaxyDB。

根据 ICMIP 的系统设计原理, 通过 Java ORB 将关系数据库系统 GalaxyDB 与 Web 集成为 ICMIP 的一个信息管理构件, 其公共接口用 IDL 语言描述如下:

```
module Galaxydb {
    interface ExecObj {
        void Initdb( in string dbname );
        string ExecCmd( in string command ); };
    interface QueryObj {
        string ResultStatus( );
        string FetchValue ( in short TupNum , in short FieldNum );
        short ntuples( ); };
};
```

Galaxydb 模块定义了两个 IDL 接口: ExecObj 和 QueryObj; 其中 ExecObj 接口用来初始化数据库和提交 SQL 语句, 并且向客户端返回一个实现了 QueryObj 接口的对象, 该对象含有数据库服务器执行 SQL 语句后返回的结果集; 用户可以根据该对象提供的 ResultStatus、FetchValue、ntuples 等方法对结果集进行相应的操作。

5 结束语

在当今的网络信息时代, CORBA / IIOP 已经成为实现分布对象互操作的技术标准。ICMIP 不仅提供了数据库 SQL 服务, 支持信息管理, 而且用户可以采用标准的 CORBA 技术对遗留系统进行集成。

在应用单位的大力支持和直接参与下, 本平台已经在湖南省通信技术开发公司开发的“电信运行维护管理系统 OMA200”、北京系统工程研究所开发的“基于 Internet 的模糊分类和检索系统 GD”, 以及航天工业总公司二院二十三所和国防科技大学 CAD 中心联合开发的“电子设备并行设计环境 YC-bridge”等项目中获得应用, 并已经取得了良好的经济效益和社会效益。

参考文献

- 1 谭永东, 何红等. 基于分布式对象的 Internet 系统结构. PCWORLD CHINA 1997. 7
- 2 吴泉源, 王怀民, 邹鹏. YHCS 客户/服务器计算机系统研究. 计算机学报, 1997, 20(增刊) 4- 9