

CORBA 在网络管理系统中的应用 Application of CORBA in Network Management System

蔡启先
Cai Qixian

(广西工学院计算机工程系 柳州 545006)

(Department of Computer Engineering, Guangxi University of Technology, Liuzhou, 545006)

摘要 提出利用 CORBA 对象与语言及平台无关的特性,将 CORBA 作为网络管理系统和被管理系统之间的通信中间件,构建一个三层结构模式的网络管理系统框架,并用于实际。

关键词 网络管理系统 CORBA 中间件

中图法分类号 TP311.13 6

Abstract A three-level framework of network management system is developed. The CORBA is used as communication middleware between the network management system and the system managed based on the CORBA objects being independence to language and platform.

Key words network management system, CORBA, middleware

现代网络管理系统一般分为 2 种,一种是基于 TCP/IP 协议簇的简单网络管理协议(Simple Network Management Protocol,简称 SNMP)体系结构,一种是基于 OSI 协议簇的公共管理信息协议(Common Management Information Protocol,简称 CMIP)体系结构。由于众所周知的原因,实时性的 TCP/IP 协议标准是当前互联网上的主流协议,而建立于其上的 SNMP 也以其简单性和灵活性战胜了 CMIP 成为当今广泛使用的工业标准,并正被积极推广至其它协议栈。然而,尽管 SNMP 有组织简单、使用灵活的优点,但它的自身缺点也不容忽视,表现在:(1)集中式管理很难适应大规模网络管理的需要,高度集中的管理造成工作站及其周围网络负荷越来越大,性能越来越差。(2)对网络动态变化支持得不够^[1]。

目前,典型的数据网络的平均无故障时间(MTTF)小于 25 d,60%的网络失效持续 10 min 至几个小时,其可信性远远低于 PSTN 网络^[2]。因此,要提高网络的效率,除了加大网络带宽外,必须寻求高性能的网络管理系统 NMS(Network Management System)。对此,有效的解决方案是在运行系统的开发过程中引入 CORBA 平台作为中间件。CORBA 可以提供面向对象的分布式应用。在开发分布式系统时,首选的技术就是 Microsoft 的 COM/DCOM 和 OMG 的

CORBA。而按照 CORBA 规范的规定,完全可以做到 COM/DCOM 和 CORBA 之间的互操作,因此从长远的角度来看,选用一种独立于具体厂家的技术即 CORBA 来开发多厂家环境下的网管系统更为合理。另外,CORBA 的透明调用机制则对系统的伸缩及软件升级提供了良好支持,能有效地适应网络的动态变化^[3]。

1 基于 CORBA 技术的实用化网络管理系统

与 SNMP 相比,CORBA 可以保证 NMS 跨多个网络管理平台,同时提供一系列可以重复使用的有关被管理对象的类,而且可以集成不同的网络管理协议。不过,要将 CORBA 应用到网络管理系统,必须考虑以 CORBA 为中心的运行系统的开发的框架与基于已有标准的管理系统的交互和系统管理等问题。网络管理论坛(NMF)制定的 CORBA 和 SNMP、CMIP 域之间的管理规范 JIDM,目的是解决如何使用 CORBA 建立管理应用和被管理应用^[2]。因此,要想建立以 CORBA 为基础的网管体系,就必须遵照网管系统的管理功能,对现有的通用 CORBA 对象服务进行完全继承或者部分改造,甚至重新定义新的服务,从而构建适用于网络管理域的对象框架,以满足网络管理业务的要求。

尽管 CORBA 代表了分布式计算技术的未来,在网络管理系统的开发、建设中具有广阔的前景,但是,在实现以 CORBA 为基础的网管系统产品时,需要为分布式处理系统提供一个有效的环境,其代价相当昂贵。因此,在目前的情况下,CORBA 比较适于作为一种数据通信接口来使用。通过这通信接口,所有的网管数据被收集给管理者,然后再对这些数据进行分析、处理。在 OMG 已经定义的 18 种 CORBA 对象服务中,目前可以在网管系统中充分利用的有:管理对象命名服务、管理对象生命周期服务和事件服务。

基于以上的讨论,本文提出了一种基于 CORBA 技术的实用化网络管理系统框架,如图 1 所示。

在该框架结构中,CORBA 服务在网络管理系统和被管理系统之间作为中间件,整个 NMS 完全以 CORBA 的体系结构为基础,通过 ORB 进行通讯就改变 NMS 传统的体系结构为完全的分布式结构。和 SNMP 相比,CORBA 可以保证 NMS 跨多个网络管理平台,同时提供一系列可以重复使用的有关被管理对象的类,而且可以集成不同的网络管理协议。对于网元层提供的接口仍采用现有的接口标准,如 SNMP 等;网络管理系统中的数据库采用成熟的 SQL 接口;在 GUI 和运行系统之间采用 Java/Web 技术,能够基于不同平台开发统一的图形用户界面,实现管理业务的灵活接入。

CORBA 可以作为实现被管系统的通信接口。在采用 CORBA 对象描述管理对象的情况下,可以将每一个管理对象类定义为一个 IDL 接口^[4]。因此,对所有基于 CORBA 的管理系统,有必要定义一个通用的管理对象。其中,一个基本管理对象接口能够包括像管理对象的类、管理对象实

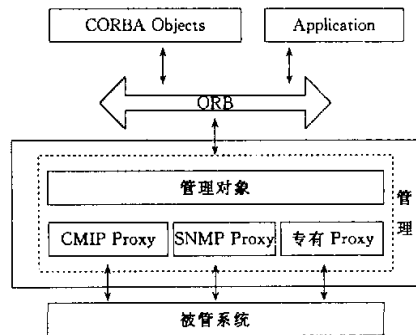


图 1 基于 CORBA 技术的实用化网管系统

例的标识、管理对象的特性等信息,基本管理对象类型接口能够被系统中每一个管理对象具体化,管理系统可以使用 CORBA 的静态和动态接口来获得被管对象提供的服务。

2 系统实现

本系统实验环境以 Jbuilder 5.0 为平台,利用本学院的校园网环境,软件开发语言为 Java2,CORBA 产品则直接利用 Jbuilder 5.0 自带的 Visibroker,下面介绍主要的编程模块。

(1)IDL 对服务对象的描述。

```
module EquipReg
{
    interface EquipRegister
    {
        void EquipLogin(in string Name,in string IpAddress);
        //UserName 为用户名;IpAddress 为被管设备的 IP 地址;
        //EquipLogin 方法用于由中央管理结点登记接入网络时的被管设备;
        void EquipLogout(in string Name,in string IpAddress);}}
    //EquipLogout 方法用于被管设备撤出网络时告知中央管理结点,请求注销;
```

(2)生成客户端和服务框架。

利用 VisiBroker 中的 IDL 编译器 idl2java 中的命令:idl2java EquipRegister.idl,生成实现该接口的 Java 代码并创建 1 个名为 EquipReg 的子目录(与 EquipReg Java 包相对应)。该目录内有 8 个文件,即 8 个 Java 类,包括 1 个接口、1 个 stub、1 个 skeleton 和几个帮助类,从而构建一个框架。在此基础上可用 Java 创建客户机/服务器模式的 CORBA 应用程序。

(3)构建服务器方。

在生成服务器框架类后,作为服务器对象的 EquipRegister 对象,要向服务器 ORB 进行登记。为此可设计一个服务器应用程序,作用是让服务器 ORB 登记该对象,并用这个新对象来扩充 skeleton class,实现 ORBInfo 接口。下面是建立服务器类的程序段:

```
public class Server
{
    public static void main(String[] args)
    {
        try
        {
            org.omg.CORBA.ORB GXorb = org.omg.CORBA.ORB.init(); //初始化 ORB
            org.omg.CORBA.BOA GXboa = orb.BOA—init(); //初始化 BOA
            //用"EquipRegister"向 ORB 登记类,客户机利用这个字符串来检索对象。
            ORBQuery serverQuery = new ORBQuery("EquipRegister");
            //调用 GXboa.obj_is_ready(),通知 ORB 准备就绪,输出新产生的对象
            GXboa.obj_is_ready(serverQuery);
            System.out.println(serverQuery+" is ready.");
            GXboa.impl_is_ready();} // 等待输入请求
            catch(org.omg.CORBA.SystemException e)
            { System.err.println(e);}}
```

(4)客户端应用。

客户端应用程序同样需要向客户机 ORB 登记。当要获得远程 CORBA 对象时,客户机采

(下转第 243 页)

2.4 添加、修改或禁用订阅服务器

展开分发服务器 WEB_SERVER, 单击“配置发布、订阅服务器和分发”按钮。

若要启用订阅服务器, 请单击“订阅服务器”选项卡, 请选择订阅服务器。

如果订阅服务器没有列出, 要想启用它, 请在“订阅服务器”选项卡上, 单击“新建订阅服务器”, 选择 SQL Server 数据库, 然后输入服务器连接信息。

若要禁用订阅服务器, 请单击“订阅服务器”选项卡, 然后清除订阅服务器旁边的框。

3 结束语

复制配置完成后, 院外门诊的服务器通过互联网, 拨号上网, 在 SQL Server 企业管理器中启动同步代理, 系统自动同步了院内服务器和院外门诊服务器的调价、收费信息, 提高了医院的工作效率, 同时医院的调价信息和院外门诊的财务信息得到及时反应和监督。

参考文献

- 1 吴晓龙. 高可用性集群系统在医院信息系统中的应用. 广西科学院学报, 2002, 18(1): 19~21.

(责任编辑: 黎贞崇)

(上接第 237 页)

用了一种间接的方法, 它通知客户机 ORB 其意图, 由 ORB 负责 ORB 到 ORB 的通讯。这种请求方式由下面 2 行代码实现:

```
org.omg.CORBA.ORB GXorb = org.omg.CORBA.ORB.init(this); // 初始化 ORB  
// 检索要调用的 EquipRegister 接口对象
```

```
ALLEquipRegister = EquipReg. NameRegisterHelper.bind(orb, "EquipRegister");
```

执行了 bind() 方法调用之后, ALLEquipRegister 本地变量与服务器的 EquipRegister 对象绑在一起。这一操作完成之后, 就可以调用帮助方法来实现客户端的应用了。

3 结束语

CORBA 作为一种规范, 为分布式处理提出了十分完美的解决方案。但是, 实际的 CORBA 产品还存在许多缺陷, 例如不同厂家的 ORB 之间的互连问题。即使同一厂家的产品, 在 CORBA 规范的符合性方面也存在着问题。因此, 在计算机网络的实际环境下, 真正实现以 CORBA 为基础的网络系统, 还有很多工作有待完成。

参考文献

- 1 The Object Management Group. CORBA-based telecommunication network management system, 1996. 5.
- 2 李影. 网络管理综述. 计算机世界, 1999. 28.
- 3 曹晓阳, 刘锦德. CORBA 服务集综述. 计算机科学, 2001, 28(3): 7~10.
- 4 李静, 李平均, 李增智. 利用 Java 实现基于 CORBA 的计算机网络管理. 计算机工程与应用, 2000, 14: 51~53.

(责任编辑: 黎贞崇)