

李永强 杨林楠 何汉明 (云南农业大学
650201)

摘要: 本文介绍了多层分布式应用系统和CORBA的基本概念,分析了CORBA在分布式应用系统中的作用及其结构,结合开发工具Delphi给出了开发基于CORBA的分布式应用系统的一般方法和步骤。

关键词: CORBA 多层分布式 对象信道

基于CORBA的多层分布式应用系统的开发

Development of Multi-Tier Distributed Application System Based on CORBA

1 引言

随着Internet/Intranet的普及应用,特别是COM/DCOM技术和各种中介软件的出现,开发基于COM/DCOM的多层分布式应用系统也逐渐流行起来。但是这些多层应用系统都是局限于Microsoft的窗口平台,对于一个拥有UNIX, Windows和大型主机,并且执行着大量关键任务的企业环境,如金融系统,通信系统等,多层应用系统必须能够整合这些不同的平台,而且要保证这些关键任务能够安全高效的运行,否则多层应用系统的实用性将大打折扣。要开发这样的系统,使用目前工业标准的分布式对象技术CORBA是最好的选择。

2 CORBA 的结构

2.1 基本概念

CORBA是Common Object Request Broker Architecture的缩写,它是由OMG定义的分布式对象标准结构。CORBA使用了面向对象的设计结构,允许软件对象在不同的操作系统平台和应用程序之中重复使用。

2.2 CORBA 的通信结构

在CORBA的环境中,应用系统主要分为两个部分。第一个部分是中介的CORBA对象,这些CORBA对象由程序员编写并且执行

在中介的应用程序服务器中。而第二个部分就是用户使用的客户端应用程序了,这些客户端应用程序通过CORBA技术使用应用程序服务器之中的CORBA对象提供的服务来执行其工作。

那么客户端应用程序和中介的CORBA对象是如何通信的呢?基本上当客户端应用程序调用中介CORBA对象时,是通过Object Request Broker连接到中介的CORBA对象,图1说明了客户端, ORB, 以及CORBA对象的通信结构。

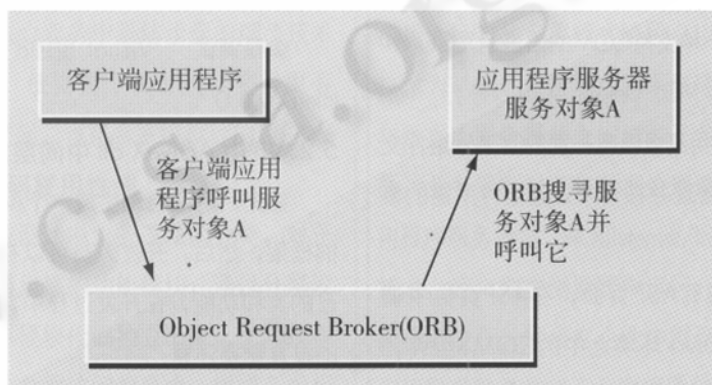


图1 客户端应用程序, ORB, CORBA 对象间的关系

在CORBA环境中,客户端应用程序并不需要知道中介的CORBA对象存在于什么地方,这些对象可能在本地的机器中,或是存在于远程的机器中。不管CORBA对象在局域网络或是广域网络中的任何机器中,客户端应用程序只需要知道需要哪一个CORBA对象提供服务,ORG便可以在局域网络或是广域网络中找到这个CORBA对象,传递客户端应用程序的要求,并且负责把执行结果回传给客户端应用程序。

CORBA对象是一个提供特定服务的对象,它通过定义一组输出接口来声明这个CORBA对象可以提供的服务。在这个输出接口中,CORBA对象定义了客户端应用程序可以调用的方法以及存取的属性。因此客户端应用程序只需要指明提供服务的CORBA对象名称以及要调用的方法或是要存取的属性就可以通过ORB得到CORBA对象提供的服务。

3 CORBA 的服务

在CORBA中,是以各种不同的软件元素来提供一个分布式应用系统所需要的各种功能,这些软件元素在CORBA之中称为服务。在CORBA的功能规格中OMG定义了许多不同目的的服务,各个都有不同的功能。在使用CORBA开发分布式应用系统时,可以根据应用系统的需要加入相对应的服务,下面是开发实际的分布式应用系统时需要使用的服务。

3.1 Event Service

CORBA的Event Service提供了分布式对象之间标准的通信机制。Event Service以“提供者-消费者 (Supplier-Consumer)”模型并且通过一个事件信道让提供者对象以异步方式传递事件给消费对象。提供者对象可以传递任何的事件给消费对象,例如某笔数据被删除或是修改了,或是库存量已经低于安全的存量了。在Event Service中,消费者对象也可以同时是提供者对象,反之提供者对象也可以是其他事件的消费者对象。

3.2 Naming Service

在多层分布式应用系统中,一般都包含了许多企业对象。如何为这些企业对象命名呢?CORBA的Naming Service提供了在分布式应用系统中为企业对象命名有意义且独特名称的服务。Naming Service允许使用阶层式的方式为企业对象命名,可以在程序代码中使用这些有意义的名称来存取这些企业对象。

3.3 Transaction service

在传统的应用系统中,要进行数据的Two-Phase Commit交易管理时,是直接程序代码中调用交易管理的API,并且把所有需要执行的工作包裹在交易管理API之中。但是在分布式应用系统中,一个交易管理的行动可能包含了数个分散的企业对象。这些参与交易的企业对象可能在不同的机器中执行,这和传统的应用系统执行交易管理的行为是完全不一样的。CORBA的Transaction service就是提供分布式应用系统能够执行关键性任务,并且允许企业对象在异质平台和数据库

之间进行数据的Two-Phase Commit交易管理工作,Transaction service允许分布式企业对象在Internet/Intranet环境中参与数据的交易管理。

3.4 Security Service

CORBA的安全服务包括了存取控制 (Access Control), 稽查控制 (Auditing), 认证控制 (Authentication) 以及存取策略控制 (Policy Control), 可以在分布式应用系统中实现非常严格的安全控管的系统。一般来说CORBA的Security Service会使用操作系统本身对于客户端用户的安全控管功能,例如登陆控制以及用户能够存取的系统对象权限等。

4 CORBA 分布式应用系统开发方法

在使用CORBA开发分布式应用系统时,必须先开发中介的CORBA企业对象。在Delphi中可以使用可视化Type Library编辑器来定义CORBA对象包含的接口,方法以及属性。不需要直接编写IDL,而由系统自动完成IDL原始程序档案的生成。

4.1 开发 CORBA 应用程序服务器

(1) 启动Delphi5, 然后点击File|New... 开启New Items对话框。选择建立Console Application图标以便建立一个在Dos Box中执行的应用程序。

(2) 选择Multitier页次中的CORBA Object图标以便在这个Console应用程序中建立一个CORBA对象。在打开的“CORBA Object Wizard”对话框中填入CORBA对象的名称以及执行实例和使用的线程模型信息。

(3) 在CORBA对象对话框中点击OK按钮,将出现Type Library编辑器以供定义这个CORBA对象提供的服务。如图2定义一个ShowMessage方法让客户端应用程序调用并且显示一个消息。

在定义完成CORBA的对象方法后,要为其加入相应代码,如:
procedure TcoShowMessage.
Showmessage;

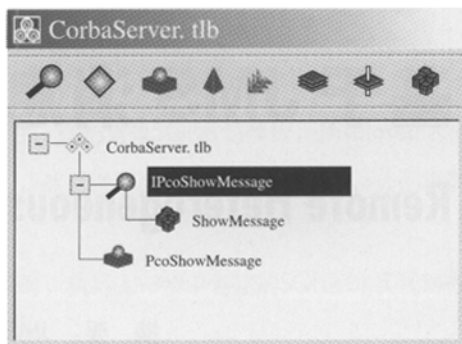


图 2

begin

writeln('这是基于CORBA建立的应用程序服务器程序');

end;

(4) 编译上面建立的CORBA对象,若没有错误,执行Vbroker \ Bin目录下的OSAbent应用程序激活VisiBroker的Smart Agent,以便客户端应用程序能够连接这个CORBA对象并且使用它所提供的服务。最后,执行这个CORBA应用程序服务器。至此,一个简单的CORBA应用程序服务器即构造完成。

4.2 客户端 CORBA 应用程序的开发

(1) 在Delphi的集成开发环境中建立一个新的Console程序,Delphi会产生下面的骨架程序代码:

```
program Project1;
{APPTYPE CONSOLE}
uses SysUtils;
begin
    //Insert user code here
end.
```

由于客户端应用程序需要使用CORBA的功能,因此应在uses部分加入使用CorbaObj以及前面coShowMessage的Type Library Wrapper程序代码:

```
uses
    CorbaObj,
    CorbaServer_TLB,
    SysUtils
```

(2) 连接CORBA服务器, 建立其中的coShowMessage对象, 再调用它的ShowMessage方法。

要连接CORBA服务器, 客户端应用程序首先必须调用CorbalInitialize, 以便初始化CORBA执行期函数。在初始化CORBA之后, 便可以使用定义在CorbaServer_TLB之中可以建立coShowMessage对象的工厂类别TcoShowMessageCorbaFactory在服务器应用程序中建立一个远程coShowMessage对象并且连接这个远程的coShowMessage对象。TcoShowMessageCorbaFactory的CreateInstance方法可以建立一个coShowMessage对象, 它接受一个参数, 这个参数是指明客户端应用程序要连接的CORBA对象的名称。如果客户端应用程序只需要连接任何的coShowMessage对象, 而不需要连接到特定的coShowMessage对象的话, 则只需要传递空字符串给CreateInstance即可。下面是客户端应用程序的主体程序代

码, 它使用了上面说明的步骤来连接远程CORBA对象并且调用coShowMessage的ShowMessage方法:

```
var
    coObject:IcoShowMessage;
begin
    //Insert user code here
    CorbalInitialize;
    try
        CoObject:=TcoShowMessageCorbaFactory.CreateInstance("");
        CoObject.ShowMessage;
    except
    on E:Exception do writeln(E.Message);
    end;
end.
```

(3) 执行Vbroker\Bin目录下的OsFind这个应用程序。OsFind能够搜寻网络中所有的SmartAgent以及目前能够提供服务的CORBA对象。

(4) 编译执行客户端应用程序, 则其将调用前面开发的CORBA服务器之中的

coShowMessage对象的ShowMessage方法以传递一信息到客户端。

5 结束语

我们可以在不同的机器中分别执行应用程序、服务器程序和客户端应用程序, 通过长期的开发应用表明CORBA在远程机器中的表现比DCOM方便很多, 不管CORBA对象是在网络中的那一台机器执行, 客户端应用程序都可以调用到它。不需要像DCOM那样需要在客户端和服务端先注册Proxy/Stub DLL或是Type Library。

参考文献

- 1 李维, Delphi5.x分布式多层应用系统篇, 机械工业出版社, 2000。
- 2 吕宗智、汪世攀、王晟, Delphi实用技术精粹, 人民邮电出版社, 2000。
- 3 王涛, 多层分布式数据库实战, 清华大学出版社, 2000。