维普资讯 http://www.cqvip.com

CORBA 与 EJB 集成技术的研究*)

The Integration Study of CORBA and EJB

李 磊 王怀民

(国防科技大学计算机学院 长沙410073)

Abstract As the server side component model, EJB is now widely used in the distrbuted application. In order to integrate with the legacy applications, EJB must integrate with CORBA sysyem. At the same time, it is necessary for CORBA applications which simplify the middle tier server development to integrate with EJB. In this paper, we primarily illustrate the integration of CORBA products and J2EE EJB through Name Server bridging on the basis of introducing the EJB and CORBA technology. The same methods can be used as patterns to integrate with other EJB products, so can be used for other CORBA products and EJB.

Keywords Integration, CORBA, EJB, RMI—IIOP

1 引言

随着分布计算技术的发展,Sun 的 Java EJB 技术、OMG 的 CORBA 技术、微软的 COM+技术开始成为分布应用的主流平台,其中,EJB(Enterprise JavaBeans)[1]是开发和配置面向对象的分布式企业级 Java 应用的服务器端构件模型,它可以简化服务器端的应用开发,使开发者可以专注于应用的业务逻辑,同时这些应用可以一次开发并装配到任何支持 EJB 的服务平台上,可以提高应用开发的效率。

CORBA(Common Object Request Broker Architecture)[2],是由 OMG(对象管理组织)制定的著名的分布对象计算规范,主要解决异构平台上分布对象的互操作和可移植问题,其核心机制是对象请求代理(Object Request Broker),简称 ORB,目前,OMG 的 CORBA 是工业界支持最多的分布计算标准。

COM+[3]技术是微软提出的分布网络计算体系 DNA 的核心,通过 COM 规范以及 DCOM,MTS Server,微软支持的 COM+在其自身的操作系统平台上已得到了很好的应用。

由于 EJB 和 CORBA 由众多的厂商支持,因此比 COM+有更加广阔的发展前景。在分布式应用开发中,EJB 和 CORBA 作为分布式对象编程模型,具有各自的特点,其互补性强。对于 EJB,利用 CORBA 的跨平台多语言特性,可以很好地提供对遗留应用的封装;对于 CORBA,可以利用 EJB 简化服务器端的开发,增强复用,缩短开发周期,同时不失 CORBA 的跨平台特性。

本文所讨论的内容是,在现有符合规范的 EJB 和 CORBA 的产品基础上实现的 CORBA 与 EJB"无缝"集成方案,它使 CORBA 与 EJB 在各自领域的优点得以合并,从而使相关应用需求的解决成为可能。

2 CORBA 与 EJB

2.1 CORBA

基于 CORBA 的分布式应用的最大特点是接口和实现相分离,CORBA 将客户方和服务器方需要彼此之间相互通讯的基本信息封装在接口(interface)中,接口定义了对象实体的属性和行为,包括服务方对外提供、客户方将要请求的各种方法的定义,为描述接口,CORBA 使用了中性语言——接口定义语言(Interface Definition Language,IDL),通过 IDL 编译器 IDL 语言向 C++、Java 等实现语言的映射,生成客户方的 stubs 接口程序和实现方的 skeletons 程序,从而使应用开发者在基于 stubs 和 skeletons 开发应用时可以不去关心网络编程的具体内容及细节,client 方只需按 IDL 文件提供的接口来调用 server 方对象实例,而 server 方只需按 IDL 文件提供的接口编写接口实现,双方都可以象开发本地应用一样开发异种平台和异种语言之间的分布式应用。

*)863项目"项目号:863-306-ZD-02基于网络的软件支撑平台与集成框架"资助。李 磊 硕士生,主要从事分布计算方面的研究。王怀民 教授,博士生导师,研究方向为分布计算环境。

维普资讯 http://www.cqvip.com

CORBA 通过对象请求代理 ORB 支持分布式环境中对象的互操作，实现对网络的屏蔽，ORB 的实现基于 GIOP(General Inter-ORB Protocol)协议，ORB 的基本功能是：当应用访问远程服务时，通过 stubs 调用 GIOP 引擎，将调用信息编码为 GIOP 字节流消息包，然后使用底层传输机制将该消息包发往服务方，并且对可能的返回值(或异常)解码；GIOP 在 TCP/IP 上的映射协议是 IIOP(Internet Inter-ORB Protocol)协议；skeletons 则在服务方正确将 CORBA 对象具体化为相应的服务后，使用方法适配机制查找出相应的方法，并解码调用参数、激活方法、编码返回值(或异常)。

CORBA 模型不仅能解决异种平台的应用集成问题，而且也能解决异种语言应用的集成问题。

2.2 EJB

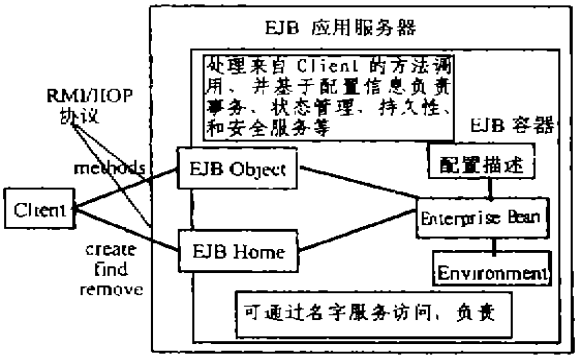

图1 EJB 的软件体系结构

EJB 技术在 Java 面向企业计算中起着重要的作用，EJB 使得开发人员开发的构件可以直接放入 Java 服务器端运行框架中执行，这些服务器端框架集中表现在容器、应用服务器，如图1所示。

Enterprise Bean(简记 Bean)配置在 EJB Server 的 EJB 容器中，容器在 Client 和 Bean 之间，起协调、管理的作用。在配置时，容器自动为每个 Bean 实例生成一个 Home 接口和 Object 接口，Home 接口对应每个 Bean 类，完成创建(create)、查找(find)和删除(remove)等管理 Bean 实例的功能；Object 接口作为 Bean 的方法的访问接口。所有 Client 发往 Home 接口和 Object 接口的请求，经容器解释并辅以生命周期管理、事务处理、状态管理、安全服务和持久服务等。

在典型的"Client＋应用服务器＋数据库服务器"三层体系结构应用中，EJB 应用服务器位于中间层，此时，EJB Bean 配置在 EJB 容器中，Client 通过容器提供的 Home 接口和 Object 接口访问容器中的 Bean，EJB 应用服务器作为 EJB 容器的容器，提供运行环境，由容器提供独立于特定业务逻辑的并发控制，事务、安全和持久性管理等服务，此处实际上是：Client 和 EJB 应用服务器以 RMI 或 IIOP 协议交互，EJB 应用服务器通过 JDBC 访问数据库。

EJB 是分布对象技术与构件技术的有效结合，但它的不足在于它完全基于 Java 平台，对于多语言多平台的遗留应用的集成，显得力不从心。

3 CORBA 与 EJB 的集成

3.1 集成方案一：Bridge

与 EJB 的集成，将使得中间层服务器的开发得以利用 EJB 构件化开发的优点，同时，客户端保留 CORBA Clients 的多语言、跨平台特性。在 CORBA 与 EJB 的集成中，必须通过以下两种机制：在开发时刻生成客户端绑定(生成对应的 stub)；在运行时进行有效的方法转换(通过 bridge)。

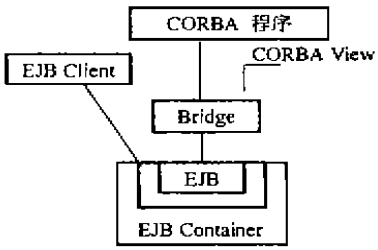

图2 CORBA 与 EJB 的集成模型

对于 CORBA 环境的开发人员，要使用 EJB 构件，首先需要将 EJB 的 Java 语言的接口定义映射为等价的 CORBA IDL 定义，然后客户端以 CORBA 调用的方式向 EJB 构件发送请求，请求被 bridge 转换后成为 EJB 的调用方法。

而对于 EJB 环境的开发人员，CORBA 的 IDL 定义首先必须映射为等价的 EJB Java 接口定义，然后 EJB 的客户端可以象调用 EJB 构件一样调用 CORBA 对象，请求被 bridge 转换后成为合法的 CORBA 调用请求。

3.2 集成方案二：RMI over IIOP

遵照图2所示，可以形成完整的集成方案，然而 Bridge 必须针对不同的构件接口单独封装，不具有通用性。

为使得 EJB(产品)实现能够直接基于 CORBA/IIOP 协议，支持与 CORBA 产品的可互操作性，OMG 和 Sun 联合定义了 EJB to CORBA 的映射规范。

为支持采用 IIOP 协议的 EJB 产品与 CORBA 互操作，Sun 引入了 RMI over IIOP，以下简记为 RMI-IIOP，并且 Sun 和 IBM 公司联合开发了 RMI-IIOP。RMI-IIOP 的实现基于 OMG 标准：Java-to-IDL mapping 和 Objects-by-Value，它在 IIOP 协议之上提供了 RMI API，通信的底层是 IIOP，RMT-IIOP 主要包括：

维普资讯 http://www.cqvip.com

1)rmic 编译器　生成客户 stubs 和服务器端的 skeletons 代理;将 Java RMI APIs 定义的对象接口,转成相应的 IDL 接口文件;

2)CORBA API 和 Java ORB　使客户程序可以访问 Enterprise Bean。

有了 RMI-IIOP,开发者可以用 JavaRMIAPIs 定义远程对象的接口,然后用 rmic 编译器生成相应 IDL 接口文件,于是,当用 RMI-IIOP 开发基于 Java 的分布式 CORBA 应用时,不需要学习 IDL 语法和编写 IDL 接口;目前,RMI-IIOP 既可以作为单独的一个软件包下载,也包含在 JDK1.3中。

RMI-IIOP 在某种程度上,是 RMI 和 CORBA 的结合,它综合了 Java 的易编程特性和 CORBA 的跨平台多语言优点,它使 EJB 和 CORBA 的集成开发成为可能。

3.3　集成方案的实现

我们的目标是在国防科大计算机学院符合 CORBA2.3规范的 StarBus 分布计算平台(ORB)下,实现 StarBus 应用(CORBA)与基于 IIOP 协议的 EJB 应用(Sun J2EE)的集成。

对 J2EE 中 EJB Bean 的 Class,利用 rmic 编译 Home 接口得到对应的 IDL 文件;根据 IDL 实现 CORBA Client(C++或 Java);在 CORBA Client 获得 Bean Home 接口的引用,然后再调用 Home 接口的 Create 或 find 方法,此时存在两个问题:

(1)对于 C++Client,要先对生成的 IDL 文件作适当修改,删除 IDL 文件中 Java 语言特有的部分;

(2)但由于 Sun RMI-IIOP 的名字服务器 tnameserv,为了兼容 JNDI 的缘故,目前并不遵从 OMG 制定的名字服务器可互操作规范(Interoperable Naming Service specification),tnameserv 采用了非标准的 Bootstrap 方法,所以即便取得其名字服务器的对象引用,也无法在 StarBus Client 中解析对象,对 Home 接口的解析,只能通过对象的字符串化的 IOR。

3.4　桥接名字服务器的实现

我们采用桥接名字服务器,以实现对 J2EE EJB 的名字服务 tnameserv 的访问桥接,我们称之为 PnameServ,提供对 Sun J2EE 中的 EJB 名字服务器 tnameserv 的 YB 接口封装服务,用于将 tnameserv 接入 YB 环境中,供普通的 YB 用户使用,提供与 StarBus 的名字服务一致的 API 接口,使得在编写 CORBA Client 时,只要取得 PnameServ 的引用,如同对 StarBus 名字服务一样解析对象,从而实现对 J2EE EJB 的透明访问。

COSS 规范中的名字服务:国际对象管理组织 OMG 在 CORBA 标准之上又定义了公共对象服务规范 COSS(common object services specification),COSS 规范包含一组公共对象服务,使得用户更方便地对对象进行操作。名字服务为 COSS 规范之一,COSS 为名字服务定义了一套完整的接口和操作,以提供对 CORBA 对象的绑定、解析等。在 CORBA 应用程序中通过 COS 名字服务程序包 org.omg.CORBA.CosNaming 访问。

名字桥接涉及以下两个主要操作:名字解析和名字绑定。前者在名字上下文中,通过对象的名字 n 查找与该名字相应的对象引用的过程称为名字解析。通过名字解析所获得的对象引用既可以是上下文对象的引用,也可以是普通对象的引用;操作为:Object resolve(in Name n)。名字绑定将名字与所指对象的对象引用关联在一起的行为称为名字绑定。一个名字绑定总是在一个名字上下文(Naming context)中的绑定,名字上下文对象对应于名字服务器接口 Naming Context,在名字上下文中包含的是名字绑定的集合,每个名字互不相同,一个名字上下文也可以被绑定到另一个名字上下文中,名字绑定包含对两类对象的绑定,即对普通对象的绑定和对名字上下文对象的绑定,对应的操作分别为 bind()和 bind context(),如下:

```
void bind(in name n,in objcet obj)
void bind_context (in name, in NamingContext nc)
```

桥接名字服务器 PnameServ 的实现遵循 CosNaming idl定义的接口,其中名字解析实现的伪码如下:

```
org.omg.Corba.Object resolve()
{
  利用Sun ORB访问 tnameserv 获得对象引用 obj1;
  通过Sun ORB的 Object_to_string()方法得到对象字符串化的IOR;
  将IOR 经 StarBus ORB 的 string_to_object()方法还原得obj2;
  Return obj2;
}
```

4　一个应用实例

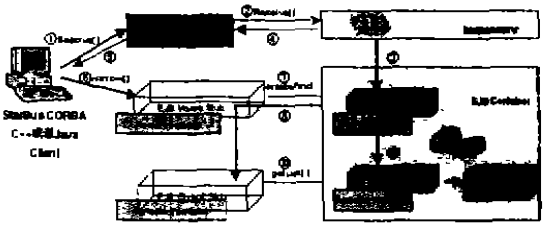

图3　CORBA Client 访问 EJB

在我们的实现中,与名字服务器相关的操作步骤是:

(下转第16页)

维普资讯 http://www.cqvip.com

数据库系统提供一个集中的地方来存储和维护数据，由数据库管理系统、数据库应用程序和数据库三大部分组成。数据库系统适于对大容量数据进行管理，并提供结构化查询语言(SQL)来方便检索。大多数网络编程语言对 SQL 语句的支持以及数据库本身对网络的支持使得数据库系统对电子商务的实现提供了有利的支持。不足的地方在于数据库系统结构的限制，开始时数据结构考虑得不完全将会给后续工作带来较大影响。

较为有效的方法是首先运用文件系统运营一段时间，待把所有要存储的数据及其结构搞清楚后，再利用数据库来提高管理效率。

②数据库的选型。数据库的选型要考虑对 Internet 计算环境和 Web 的支持力度，以及是否支持多媒体数据的存储。并发访问控制能力和联机事务处理(OLTP)是另外两个重要的考虑因素。

③表。数据库中存储的数据应包括商品信息，如商品名称；厂家信息，如厂家联系方式；配送公司信息，如送货方式。为保证数据安全，需定期备份数据库中数据。

2.3.4 服务器选型及系统结构 目前使用较多的是 NT 服务器和 UNIX 服务器，NT 服务器以其方便易用获得很多网站的青睐，但其稳定性及部分安全漏洞有待进一步完善；而 UNIX 服务器的强稳定性、可靠性和容错能力在许多大型网站得到了进一步的验证。服务器的选择要根据财力、站点规模、安全要求等因素综合考虑。

系统结构采用客户端、应用服务器和数据库服务器三层构架。

2.3.5 网络财务软件 电子商务的发展将引起财务管理的巨大变革，网络财务直接利用网络完成销售额入账、资金流量分析、财务分析、报表分析，实现网络时代财务管理的高效化。

2.3.6 市场决策支持软件 软件自动纪录顾客在网上超市的活动，自动判断用户的购买倾向，并根据各种倾向可能带来的利润，智能化地提示商家应该如何调整商务策略。用户的活动倾向数据应存入数据库，积累一定的时间，保证统计方面的合理性。

(下转第 20 页)

(上接第 29 页)

通过询问 tnameserv 所在的主机 IP 和 Port，PnameServ 即可以解析该 tnameserv 上的 EJB 对象(缺省为本机，Port 1050)，遂 PnameServ 可以在每个 CORBA Client 运行的主机上运行一个，以便于获取其引用，当然也可以共用 PnameServ，通过网络获取其引用。

PnameServ 作为 tnameserv 的桥接并不能完全取代 StarBus 的名字服务，若要访问 StarBus 的其它 CORBA 对象，仍可以用 StarBus 的名字服务。

对于 J2EE EJB 访问 StarBus CORBA Server，可以看作是 Java RMI-IIOP Client 与 CORBA Server 的关系。

由于 PnameServ 是 tnameserv 的桥接，遂可以将 StarBus CORBA 对象向 PnameServ 绑定，从而绑定到 tnameserv 上，于是 EJB 便可以访问 CORBA 对象。

小结 在现阶段，研究 EJB 与 CORBA 集成技术，提供互操作途径的研究，对于分布式环境下的应用集成，以及分布环境下构件的重用、移植、互操作，都具有重要的意义。

在具体开发中，名字服务器的管理和 Bean Jar 文件的反编译，以及对生成的 IDL 文件的修改，还缺乏一个简便的开发管理工具，我们将提供集成开发的辅助工具。

以 EJB 与 CORBA 的集成作为基础，我们将遵照 OMG 的 CCM(CORBA Component Model)标准，研究实现构件化的 CORBA 开发环境，并提供配置与管理工具。

参 考 文 献

1 Andoh A, Nash S. RMT over IIOP, Last modified April 18 2000 IBM. URL: http://www.javaworld.com/jw-12-1999/jw-12-iiop.html

2 Designing Enterprise Applications with the Java™2 Platform Enterprise Edition. March 22, 2000. the J2EE Blueprints URL http://java.sun.com/j2ee/blueprints

3 OMG. CORBA Services, Common Object Services Specification. 1998

4 http://java.sun.com/j2ee/, http://java.sun.com/rmi-iiop/

5 The Common Object Request Broker: Architecture and Specification, Revision 2.3: June, 1999