

EJB 技术及应用

钟国祥

(重庆教育学院 重庆400067)

The Technology and Application of EJB

ZHONG Guo-Xiang

(Chongqing Education College, Chongqing 400067)

Abstract Component technology has been the focus of the software industry. In this paper, the history of the component technology is reviewed briefly. The framework of EJB is introduced, its composition is mainly focused, the EJB products and its development are discussed and in the end, the conclusion is made.

Keywords Enterprise JavaBeans, Technology, Usage

1. 构件技术的发展

基于构件的技术在软件产业中是当前的热点,构件技术在软件开发上主要解决重复开发的问题。软件构件技术,厂商主要形成了三大阵营:

- OMG 阵营,它推出的跨平台、跨语言 CORBA 规范,已逐渐成为业界的标准。OMG 成员 Sun 挟 Java 利器推出的 EJB CTM,允许厂商灵活地实现事务、安全等底层服务;平台独立,实现也独立;同时给其它厂商提供了良好的发展空间。ERP、事务对象框架逐渐向 EJB 靠拢。EJB 的这些优点,众多厂商(如 BEA、IBM、Inprise、IONA、Oracle、Sun、Sybase)推出了对 EJB 支持的产品。

- DCOM 阵营,以微软为首,VC、VB、DELPHI 等都支持 DCOM,但是仍以桌面计算为基础。

- OpenDoc 阵营,以 Apple 和 IBM 为首,现在没有得到更多厂商支持。

2. EJB 模型

EJB 这种基于构件的开发模型,用于创建可伸缩、跨平台、分布式应用,并可创建具有动态扩展性的服务器应用。EJB 构件模型使得构件可包含应用系统逻辑,并且构件可重用,另外还提供资源管理、事务支持和并发性管理。EJB 是一种 C/S 结构,提供服务的方法运行在服务器端,根据客户端的请求调用相应的方法,其基本结构如图1所示。

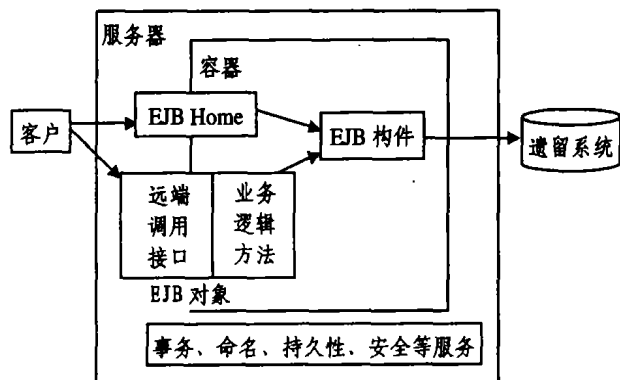


图1 EJB 模型图

2.1 客户

有两种类型客户机可以使用 EJB。一是 EJB 客户机:用

EJB API 的 Java 客户机。客户机利用 JNDI 确定对象,利用 IIOP 或 JRMP 协议上的 Java RMI 来调用远程方法;二是 CORBA 客户机:用 CORBA IDL 支持的任何语言写的客户机。客户机用 COS 命名服务来确定对象,用 CORBA IDL 来调用远程方法,用对象事务服务 OTS。

2.2 EJB 构件

EJB 中的 bean 可以分为会话 bean、实体 bean 和消息 bean。

- 会话 bean:它是短暂的对象,运行在服务器端,并执行一些应用逻辑处理,由客户端应用程序建立,其数据需要自己来管理。会话 bean 的粒度较大,可以包含很多业务逻辑,不需要反复从网络上装载。

会话 bean 可进一步分为无状态和有状态(维护客户端状态)两种。会话 bean 的配置描述符须声明该 bean 是有状态或无状态的。无状态 bean 不维护客户端状态,无状态 bean 不维护状态,可以没有需要保存的信息,容器可以删除 bean 的实例,节省了主存。有状态 bean 维护客户端状态。有状态和无状态的会话 bean 都可以访问数据库,并且参与一个事务。为了让 bean 在事务中执行它的任务,bean 开发者可以在 bean 中实现 javax.ejb.SessionSynchronization 接口,容器能自动检测这个接口,容器会使用这个接口中的方法以使 bean 得到事务的状态信息。

容器开发商可以实现把会话 bean 的实例从主存移到二级存储中,这可以增加一段时间内实例化的会话 bean 的总数。在容器维护一个 bean 的时间期限,当某个 bean 的不活动状态时间达到阈值,容器就把这个 bean 拷贝到二级存储中并从主存中删除;而当它再次活动时,就从二级存储中恢复到主存。

- 实体 bean:它是持久的对象,可以被其它对象调用。实体 bean 不维护客户端状态,且必须在建立时指定一个唯一的标识,并提供相应的机制允许客户应用程序根据实体 bean 标识来定位 beans 实例。多个用户可以并发访问实体 bean,事务间的协调由容器来完成。实体 bean 支持事务处理,当系统停机时,也可恢复。

EJB 规范中定义了两种处理实体 bean 的持久性模型,即自管理的持久性及容器管理的持久性。自管理持久性是由 EJB 自己来管理持久性,它需要 EJB 开发者来编写数据库或应用程序的处理逻辑,并加入到类的 ejbcreate()、ejbremove()、ejbfind()、ejbload()和 ejbstore()等方法中。容器管理的持

久性是将 EJB 持久性管理交给容器来完成,开发者一般要在配置描述符中的 ContainerManagedFields 属性中指定 EJB 实例持久性域。使用容器管理持久性,用户无需知道实体存储的数据源,也不需参与复杂、繁琐的编码工作。

通常实体 bean 和会话 bean 是一起使用的。会话 bean 允许保存客户端的状态信息,客户端和会话 bean 实例间是一一对应的。实体 bean 允许保存记录的信息,实体 bean 实例和记录间则是一一对应的。一个理想的情况是客户端通过会话 bean 连接服务器,然后会话 bean 通过实体 bean 访问数据库,这使得既可以保存客户端的信息又可以保存数据库记录的信息。

·消息 bean:它是 EJB2.0 对 EJB1.1 规范的一个基础性更改,专门设计用来处理 JMS 消息。JMS 是一种与厂商无关的 API,用来访问消息收发系统,并提供了与厂商无关的访问方法,以访问消息收发服务。许多厂商都支持 JMS,如 IBM 的 MQSeries、BEA 的 WebLogic JMS service 和 Progress 的 SonicMQ 等。JMS 中消息收发是异步的,即 JMS 客户机可以发送消息而不必等待回应。

EJB 系统采用两种方式使用 JMS,EJB1.1 及其以前的版本把 JMS 作为一种 bean 可用的资源,使用 JMS API 的 bean 就是消息的产生者或发送者,在这种方式下,bean 将消息发送给称为队列的虚拟通道;而在 EJB2.0 中则存在消息 bean,它是消息的使用者或接收者,它监听特定的虚拟通道,并处理发送给该通道的消息。

消息 bean 也是一种完备的企业级 bean,但和会话 bean、实体 bean 间仍存在一些重要的区别。消息 bean 没有远程接口或本地接口。这是因为消息 bean 不是 RPC 构件,它没有供 EJB 客户机调用的业务方法。同时消息 bean 类似于无状态的会话 bean,即这两种 bean 在两次请求之间都不保持任何状态。

2.3 EJB 容器

EJB 容器是 EJB 构件运行的环境,是一层代替 bean 执行相应服务的接口。EJB 容器负责提供协调管理、资源管理、版本控制、动态性、一致性、安全、事务处理和 RMI 等功能。另外容器建立上下文环境,负责切换、协调不同 EJB 对象。会话 bean 中与二级存储中交换以及实体 bean 中的持久性管理都是由容器完成的。

客户应用程序通常不和 EJB 直接打交道,而是要通过容器提供的接口:

·Home 接口。它通过称为 Home 对象的类来实现。一个 Home 对象的实例在服务器中实例化,使得客户端可以访问它们。Home 接口允许客户建立或删除 EJB。对实体 EJB 来说,Home 接口还提供定位特定 EJB 实例的功能。当用户请求 EJB 服务时,它首先要通过 JNDI 来定位对象的 Home 接口,EJB 类及其容器对用户来说是透明的。Home 接口必须提供建立 EJB 对象的方法,一旦用户找到所需的 EJB 类后,它就可以通过调用 Home 接口中的生成方法建立 EJB 对象。

·Remote 接口。它通过容器提供的 EJB 类的一个实例 EJB 对象来实现。客户总是通过调用 EJB 远调用方法来执行 EJB 服务。用户调用 EJB 对象时,由 EJB 容器接受请求,并将任务交给 EJB 对象。这种机制保证为用户及 EJB 提供透明的状态管理、事务控制及安全性服务。

2.4 遗留系统

EJB 的发展方向是要支持多种环境,包括①TP Monitor:如 IBM TXSeries(CICS&Encina),BEA Tuxedo(通过 jolt 连接);②Componet Transaction Server:如 Sybase Jaguar CTS、

Microsoft Transaction Server;③CORBA 平台:如 Inprise VisiBroker、Iona Orbix;④数据库管理系统(通过 jdbc):如 Informix、Oracle、Sybase、CloudScape;⑤Web 服务器:如 Java Web Server、Apache、Oracle Application Server。

3. EJB 角色

EJB 角色如图2所示。

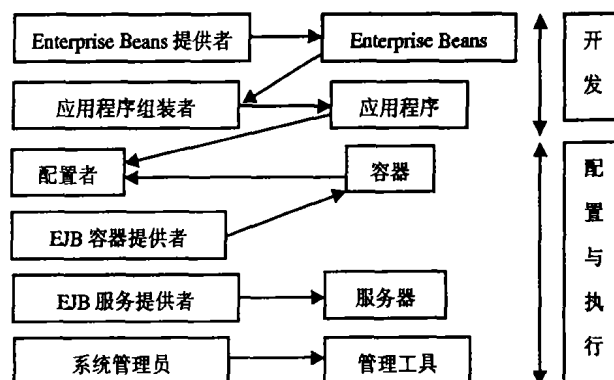


图2 角色及生命周期图

EJB 体系结构通过把编程的过程分为七个不同的角色而使开发复杂的应用系统变得简单。七个不同的角色,每个都有特定的任务:

·EJB 服务提供者:是典型的提供分布式底层服务的代理。它提供了一个分布式应用程序开发者需要的平台和设施,也提供了分布式程序的运行环境。

·EJB 容器提供者:是分布式系统、事务处理、安全方面的专家容器提供者,提供了一个 EJB 的配置工具(如 GenIC)。

·企业 beans 提供者:是金融或电信等应用领域的专家。它提供了业务方法,定义了 beans 的 Remote 接口和 Home 接口,还定义了 beans 的配置描述符。

·应用程序组装者:是一个将定制的企业 beans 和其它的构件(如 GUI 客户、Applet、Servlet 等)组装成一个完整的应用程序的家。它接受上述企业 Beans 提供者输出的 EJB-jar 文件作为自己的输入,在配置描述符中插入应用程序组装指令。组装器只关心 beans 的接口,包括 Remote 接口和 Home 接口,而不关心这些接口是怎么实现的。

·配置者:为特定的企业 beans 配置特殊的操作环境。配置者为了适应应用程序,将多个企业 beans 组合起来,通过修改企业 beans 的属性来达到配置相应的操作环境。例如,配置者通过设置配置描述符的相应的属性来设置事务和安全。配置者的另一个任务是将应用程序与现有的企业管理软件结合起来。常见的配置工具有 GenDD、DumpDD 等。

·系统管理员:管理员配置、管理企业计算和网络服务的底层,包括 EJB 服务和容器。管理员监视应用服务器的运行状况,当应用服务器不正常运行时就采取相应的措施。

·持久性管理器提供者:它是在 EJB2.0 版本中才出现,负责容器管理的持久性模型,实体 beans 持久性在容器中的安装。

传统的应用程序开发者现在是企业 beans 的提供者,也可能是应用程序的装配者。这样能使他们将主要精力放在业务逻辑上面。虽然分布式的应用程序还是比较复杂,可是应用程序设计者的工作却变得简单了,因为所有复杂的工作都交给了 EJB 服务器和容器提供者了。

4. 常见 EJB 系统

EJB 规范是 Sun 公司制定的,1.1 推出后立即引起了计算

机界的广泛关注和产业界的支持,成熟的产品如下:

4.1 WebSphere

IBM WbSphere 是一个完善的、开放的 Web 应用服务器,它是 IBM 电子商务应用架构的核心,在高级版中开始支持 EJB 的编程模型。EJB 是 WebSphere 最核心的对象技术之一,它提供了具有交易功能的服务器端的 Java 构件,同时又是一种新型的对象分布技术编程模型,使用 EJB 构件完成的应用,会支持更高级的基于数据库的交易处理功能,如多个数据库之间的更新、两阶段提交等。最关键的是,WebSphere 把 EJB 和 IBM 其它的优秀的产品(如 TXSeries)结合在了一起。在具体的应用中,基于安全的考虑,性能考虑和系统管理考虑,还可以增加 Firewall、WebSphere Performance Pack 和 Tivoli。所以,可能 IBM 的 EJB 容器不一定是最好的,但是 WebSphere 中的 EJB 和 IBM 其它产品的结合却是最好的。通过 Visual Age,可以很快地开发出 EJB 构件,然后部署在 WebSphere 应用服务器中,快速地开发一个电子商务应用。

4.2 IAS

Inprise 的 IAS 也提供了 EJB 的运行环境,不过没有和别的产品结合在一起,技术性比较浓。VisiBroker 是 IAS 的基础,所以在启动 IAS 之前,网络中至少要有一台机器运行 Smart Agent,默认端口号是:14000。IAS 是系列服务和工具的集合,通过这些服务和工具,你可以编译、配置、管理基于 Web 的多层体系结构的应用程序。IAS 中的 Visibroker 有效地实现了连接和线程的管理,如果有多个客户同时请求同一个服务器。IAS 通过连接池和线程池来有效管理这些请求。在底层,IAS 是通过 RMI/IIOP 来处理多台机器之间的通信。IAS 提供了一个完整的 EJB 容器服务,提供了一个标准的 EJB 容器工具包,这个工具包可以独立运行,也可以结合到自己的应用中去。IAS 的 EJB 容器服务可以自己创建、管理一个 EJB 容器,可以用来配置、运行 EJB,还可以监视 EJB 的运行状态。另外,IAS 也提供了自己的 JNDI 名字空间,通过 JNDI 管理 bean 的注册和资源的获取。

4.3 WebLogic

作为服务器端商务逻辑的行业标准和 Java2 平台的基石,EJB 是 BEA WebLogic 应用服务器的主要技术。早在去年 7 月,BEA 公司就宣布该公司率先实现了对 EJB 2.0 标准的支持。而 EJB 2.0 规范是目前最高的 Java 企业级开发工业标准,BEA 对 EJB 2.0 的率先支持再次巩固了该公司在 Java 应用领域无可争议的领先地位,并加强了 BEA WebLogic 作为 J2EE 技术主要平台的市场地位。BEA 是通过为 BEA WebLogic Server 5.1 提供一个附加的软件包,实现了对 EJB2.0 的支持。凭借 BEA WebLogic Server 对 EJB2.0 的支持,BEA 可帮助开发人员更快地推出电子商务应用系统,从而让开发人员极大地受益于这一业内最新标准。

5. 开发实例

本例以有状态的会话 bean 来开发简单的个人网上购物,开发工具涉及到 Jbuilder4.0,IAS:

5.1 创建 Remote 接口

```
public interface Cart extends javax.ejb.EJBObject {
    void addItem(Item item) throws java.rmi.RemoteException;
    void removeItem(Item item) throws ItemNotFoundException, java.rmi.RemoteException;
    float getTotalPrice() throws java.rmi.RemoteException;
    java.util.Enumeration getContents() throws java.rmi.RemoteException;
    void purchase() throws PurchaseProblemException, java.rmi.RemoteException;
}
```

5.2 实现 Bean 类

```
public class CartBean implements javax.ejb.SessionBean {
```

```
    public void setSessionContext ( javax.ejb.SessionContext sessionContext) {...}
    public void ejbCreate ( String cardHolderName, String creditCardNumber, Date expirationDate) {...}
    public void ejbRemove() {...}
    public void addItem(Item item) {...}
    public void removeItem(Item item) throws ItemNotFoundException {...}
    public float getTotalPrice() {...}
    public java.util.Enumeration getContents() {...}
    public void purchase() {...}
    public void ejbActivate() {...}
    public void ejbPassivate() {...}
}
```

5.3 Home Interface

```
public interface CartHome extends javax.ejb.EJBHome {
    Cart create(String cardHolderName, String creditCardNumber, java.util.Date expirationDate);
    throws java.rmi.RemoteException, javax.ejb.CreateException; }
```

5.4 Client 的实现

```
public class CartClient {...}
public static void main(String[] args) throws Exception {
    javax.naming.Context context = new javax.naming.InitialContext();
    Object ref = context.lookup("cart");
    CartHome home = (CartHome) javax.rmi.PortableRemoteObject.narrow(ref, CartHome.class);
    Cart cart;
    { String cardHolderName = "Jack B. Quick";
      String creditCardNumber = "1234-5678-9012-3456";
      Date expirationDate = new GregorianCalendar(2001, Calendar.JULY, 1).getTime();
      cart = home.create ( cardHolderName, creditCardNumber, expirationDate); }
}
```

小结 采用 EJB 可以提高开发人员的效率,构件开发人员和应用开发人员不需要实现系统中的一些复杂的逻辑结构,EJB 模型保证开发出来的构件可以被安装在任何支持 EJB 规范的系统中。

从企业应用多层结构的角度上看,EJB 是商业逻辑层的构件技术,与 JavaBeans 不同,它提供了事务处理的能力;从分布式计算的角度,EJB 像 CORBA 一样,提供了分布式技术的基础,提供了对象之间的通讯手段。从 Internet 技术的应用的角度上看,EJB 和 Servlet、JSP 一起成为新一代应用服务器的技术标准。Servlet 负责与客户端通信,访问 EJB,并把结果通过 JSP 产生页面传回客户端,成为开发的新潮流。从发展的角度看,EJB 还会进一步地完善与发展。EJB2.0 中,新的容器管理的持久性模型比以前的模型要灵活得多,它提供持久性管理器,允许各种实体为复杂的对象图建立模型;为了提供跨容器的可移植性,推出了 EJB Query Language 作为查找和选择操作的一种通用的查询语言;还在 Home 接口中增加了一些方法;改进了安全机制。另外 CCM 规范可以使得 CORBA 构件和 EJB 有效地集成起来。CCM 采用支持 EJB 规范中的对象封装方案,使 EJB 可作为 CORBA 构件重新使用,EJB 提供业务逻辑的构造能力,CORBA 则提供跨平台的分布通信能力。当然 CORBA 构件也可以在 EJB 中执行。

随着 EJB 中进程管理、线程池、并发控制和资源管理的增强,EJB 的威力正越来越大地展现出来,构成企业计算的基础。

参考文献

- 1 Monson-Haefel R. Enterprise JavaBeans Second Edition. 1999. O'Reilly & Associates, Inc.
- 2 Weblogic product introduction. Available at: <http://www.weblogic.com>
- 3 Pour G. Enterprise JavaBeans, JavaBeans & XML Expanding the Possibilities for Web-Based Enterprise Application Development. TOOLS 1999. 282~291
- 4 IAS product introduction. Available at: <http://www.borland.com/appserver>
- 5 EJB Document. Available a: <http://java.sun.com/products/ejb/docs.html>