



相关主题

RECOMMEND ARTICLE

- ▶ 基于分布式对象的网游程序结构设计(5)
- ▶ 基于分布式对象的网游程序结构设计(4)
- ▶ 基于分布式对象的网游程序结构设计(3)
- ▶ 基于分布式对象的网游程序结构设计(2)
- ▶ 基于分布式对象的网游程序结构设计(1)
- ▶ 未来游戏设计的十大技术挑战--2
- ▶ 未来游戏设计的十大技术挑战--1
- ▶ Open Inventor—Coin3D开发环境

[MORE](#)

推荐文章

RECOMMEND ARTICLE

- ▶ 数据广播方案的优化
- ▶ 网络游戏的位置同步
- ▶ 游戏音乐制作案例之《战火 红色警戒》音效制作揭秘
- ▶ 英雄连Online 原画
- ▶ 游戏音乐制作案例之《乱武天下》
- ▶ 游戏音乐制作案例之《诛仙》
- ▶ 《鹿鼎记》最新原画
- ▶ MIDP2.1规范的新特性

[MORE](#)

热门文章

HOT ARTICLE

- ▶ [电子书下载]游戏设计 — 原理与实践
- ▶ [电子书下载]网络游戏开发
- ▶ 游戏设计全过程
- ▶ [电子书下载]游戏设计技术
- ▶ [电子书下载]游戏设计理论
- ▶ CS游戏人物模型制作教程
- ▶ CG人物插画基本流程
- ▶ [转贴]MAX高级人头教程

[MORE](#)

您的位置: 开发技术



文章标题	基于分布式对象的网游程序结构设计(3)		
来源:	[srpl ab]	浏览:	[984]

虽然在游戏开发中,很少使用DCOM/COBRA分布式组件技术。但是作为一种分布式技术,这里也分析一下存在的问题。分布式组件技术是一种CS结构,其出现,是为了简化网络编程,开发者不再需要关心具体如何进行底层通信。目前比较有代表性的有两种:DCOM和COBRA。DCOM使用ORPC机制,COM服务器创建对象类的实例,一个COM对象可以具有多个接口,分别代表不同的观察角度和不同的对象行为。客户端获取对象接口的指针,通过指针调用相关的方法。

COBRA是由OMG(Object Manage Group)提出的,其核心是ORB(Object Request Broker)。作为一个透明的总线式模型,可以与本地或者远程的对象进行交互。COBRA对象对外呈现一组接口,客户端获取对象的引用,通过引用进行方法的调用。ORB负责查找对象的实现,发送请求并处理返回结果。

两种组件模型都采用CS模式的通信。为了调用一个服务,客户端需要调用远程对象实现的方法。服务器端提供的服务,封装成为一个对象,对象的接口使用IDL(Interface Definition Language)描述。客户端通过调用IDL中定义的方法与服务器端进行交互,不用关心实际对象的实现。支持面向对象的一些特性,例如:数据封装,多态,继承。COBRA支持多重继承,DCOM不支持,但是一个对象可以有多个接口。

为了调用一个远程的方法,客户端调用本地的桩函数,桩函数将参数封装成为请求,将请求发送给服务器端。服务器端将请求传递给Server Stub,对参数进行解包,并调用实际的函数。在DCOM中,客户端的桩称为proxy,服务器端的桩称为stub;在COBRA中,客户端的桩称为stub,服务器端的桩称为skelton。

假设有一种Grid对象服务,Grid对象维护一个二维的整数,支持两组方法:第一组是get和set,可以设置某个格子上的数值;第二组是reset,复位某个格子上的数值。则对于COBRA,需要定义三个接口,接口Grid1支持get和set;接口Grid2支持reset;接口Grid继承Grid1,Grid2。使用DCOM,则定义两种IGrid1和IGrid2。

采用DCOM调用,则客户端端的处理步骤如下:

1. 客户端调用COM库函数CoCreateInstance,使用CLSID_Grid和IID_IGrid1作为参数。
2. COM库请求服务器创建对象
3. 服务器端COM获取指向CLSID_Grid的类工厂指针,调用其CreateInstance函数。
4. 类工厂创建一个对象,然后服务器端调用QueryInterface获取指向IID_IGrid1接口的指针。
5. COM库将指向IGrid1接口的指针返回给客户。
6. 客户端调用IGrid1接口的get方法。

采用COBRA调用,则客户端端的处理步骤如下:

1. 客户端调用桩函数grid::_bind()。
2. ORB向Server发起请求
3. 服务器端创建实例,调用CORBA::BOA::impl::_is_ready(),通知ORB对象已经准备好。
4. ORB向客户端返回对象的引用
5. 客户端调用对象的方法。

在上面的处理步骤中每个步骤,COM库或者ORB都需要进行很多操作,因此调用的效率不高,只能够进行粗粒度的调用。例如一个服务上有几十个对象,如果通过上述方法获取对象的内容并进行显示,处理和流程将会多么复杂。

[附图见附件文档]

对象引用在客户端应用中创建,并且是动态创建的,因此当服务器端组件是细粒度,对象很多时,这种代价是非常高的。相比Web Service,WSDL描述的服务方法参数和返回值,类似于组件对外的接口,但是在进行Web Service进行调用时,不需要创建对象,也没有对象的概念。而是直接使用URL,定位到了需要调用的服务器端的对象。因此,Web Service与组件模型相比,调用需要的交互要少一些。

在DCOM和COBRA中,由于对象的创建导致交互很多,不适合于细粒度的模型。但是如果将服务器端组件的对象预先创建好,并引入对象管理功能,在客户端访问服务器端时,一次性的在客户端创建相应的对象,则可以解决这个问题,如下图:

DCOM和COBRA在开发和部署上都相对复杂,影响了其发展和应用。

http://bbs.ogdev.net/Public/WebUsers/ArticlePic/200711091652392995320hiovasf2bg.doc

本栏目登载此文出于传递信息之目的,如有任何的问题请及时和我们联系!

无任何评论!

请您注意:

- 尊重网上道德,遵守中华人民共和国有关法律法规
- 尊重网上道德,遵守中华人民共和国的各项有关法律法规
- 承担一切因您的行为而直接或间接导致的民事或刑事法律责任

发表评论:

昵称:

联系EMAIL:

- 中国网游研发中心新闻留言板管理人员有权保留或删除其管辖留言中的任意内容
- 您在中国网游研发中心留言板发表的作品，中国网游研发中心有权在网站内转载或引用
- 参与本留言即表明您已经阅读并接受上述条款

[关于我们](#) - [免责声明](#) - [联络热线](#) - [申请链接](#) - [站点地图](#) - [网站帮助](#)

Copyright © 2004-2007 盛趣信息技术（上海）有限公司 All rights reserved.
OGDEV.NET -- 网络游戏研发网 最佳分辨率 1024×768