北京郵電大學

分布计算环境

北京邮电大学计算机学院

北京郵電大學

Chapter 3

面向对象的分布计算环境

主要内容

- 分布式系统中的面向对象技术
- **CORBA技术**

分布式系统中的面向对象技术

- 传统的面向对象技术
- 分布式系统对传统对象模型的影响
- 实现分布式对象模型的机制

面向对象技术

面向对象技术已在软件生存期的各个阶段取代传统的结构化方法，成为当前软件开发的主流技术。

- **OO的基本思想**

  从现实世界中客观存在的事务（即对象）出发来构造软件系统，并在系统构造中尽可能利用人类的自然思维方式。

  程序＝对象＋相互关系

- 重要特征：封装、继承和多态。面向对象的开发过程实际上是一个建模过程。强调的是复用。
- 先进的分布式软件体系结构必须与面向对象技术结合在一起，从而可分享面向对象技术带来的众多好处。

分布式系统中的面向对象技术

- 传统的面向对象技术
- 分布式系统对传统对象模型的影响
- 实现分布式对象模型的机制

分布式系统对传统对象模型的影响

传统的对象与访问该对象的程序只能存在于同一进程中,客户进程不可能直接访问异地服务进程中的常规对象

- 传统对象的关注点：封装和通过继承对实现进行重用
  - 封装提供了一种将对象实现细节与其它对象屏蔽开的严格方法，可以大大缓解在面向过程系统中较突出的维护问题
  - 继承提供了一种重用对象实现的简便方法
- 分布式环境要求更好的可插入性：
  - 不太关注于直接重用代码，而是要求能够利用远程所实现的服务
  - 要求另一层次上的封装，即只需暴露公用接口
- 对象 ==〉 组件

分布式对封装性的影响

- 在分布式对象系统中，对象不仅要屏蔽有关的算法和数据结构，还需要屏蔽“系统是分布的”这一特性，提供分布透明性。
  - 位置透明性：用户不必关心对象位于何处
  - 访问透明性：可用一致的方式访问不同类型的机器上的对象
  - 持久透明性：对象所处的状态既可以是活动的，也可以是静止的
  - 重置透明性：对象的位置可以变化而不影响对它的调用
  - 迁移透明性：系统内部可以迁移对象的位置
  - 失败透明性：屏蔽被访问对象的失败及恢复过程 （容错）
  - 事务处理透明性：与事务处理相关的调度、监控和恢复是透明的
  - 复制透明性：用户不知道有多少个对象副本存在

分布式对继承性的影响

- 问题：分布对象计算系统中都不支持跨站点的继承性，因为实现代价太大
- 解决：把对象分为接口和对象实现。接口由接口定义语言IDL来描述，对象实现的方式则根据具体的编程语言而定
  - 接口：描述使用该对象的方法。由该对象所能提供的操作的说明组成
  - 对象实现：实际构成该对象所提供的服务。它定义了与对象有关的数据的格式和用于管理这些数据的服务。
- 分布式对象继承机制的实现：从接口继承出发

分布式对对象引用的影响

- 对象引用用于在系统中标识一个对象，客户可以根据它所知道的对象引用来与相应的对象建立连接，从而访问该对象
- 集中式系统中：对象引用由指向该对象所在的内存地址的指针来表示
- 分布式系统中：对象引用需要由一个较复杂的数据结构来表示。需要充分考虑到由系统的分布所带来的一些问题，如网络协议、网络地址和对象的迁移等
- 此外，客户端如何如何获取远端对象的对象引用？

a Remote object reference may be

32 bits	*32 bits*	*32 bits*	*32 bits*	
Internet address	port number	time	object number	interface of remote object

- a remote object reference must be unique in the distributed system.
- the first two fields locate the object unless migration or re-activation in a new process can happen
- the fourth field identifies the object within the process
- its interface tells the receiver what methods it has (e.g. class *Method*)

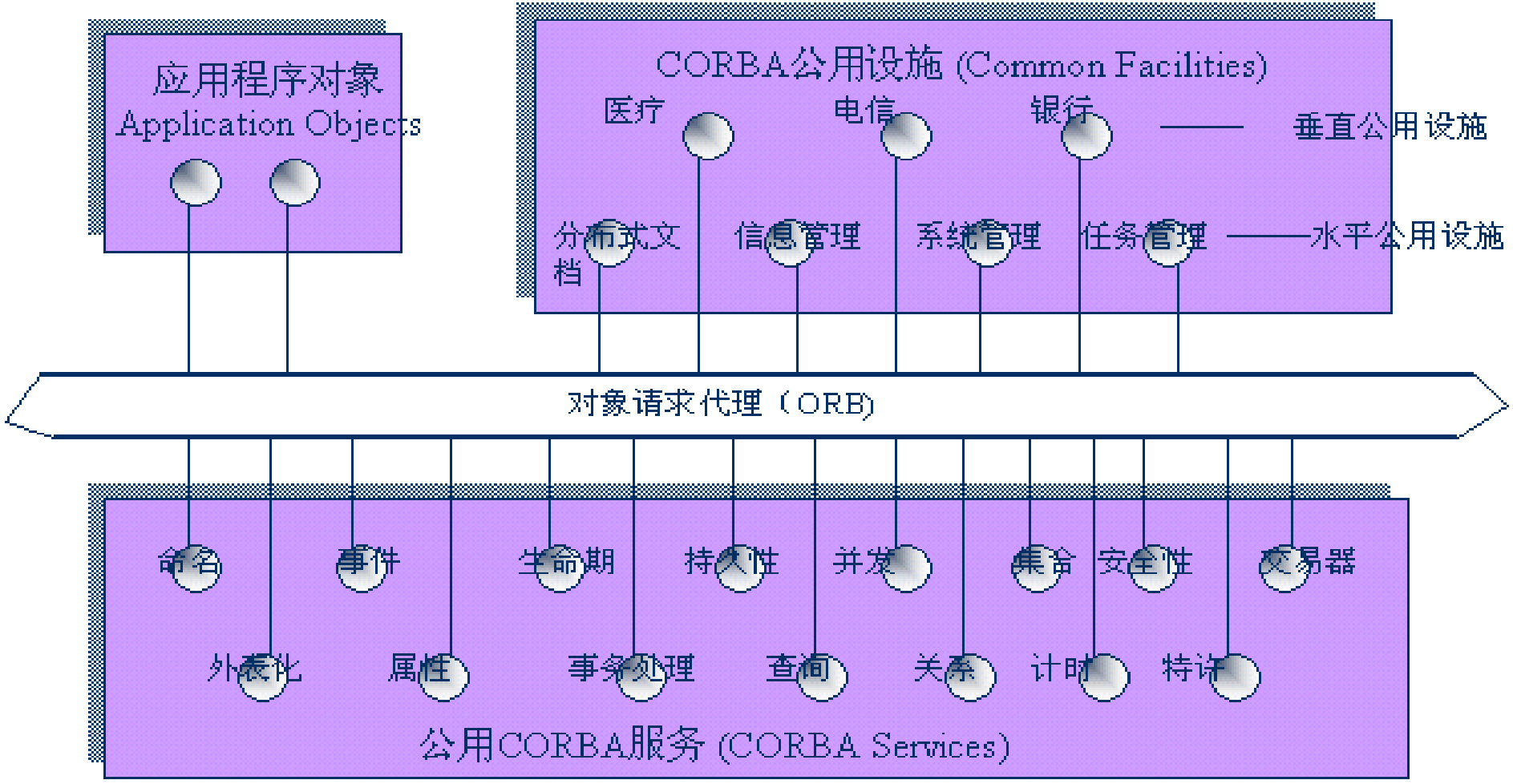
分布式系统中的面向对象技术

- 传统的面向对象技术
- 分布式系统对传统对象模型的影响
- 实现分布式对象模型的机制

实现分布式对象模型的机制

- 分布式调用：通过把相应的请求信息传递给远端对象来完成，提供位置透明性和访问透明性
- 分布式系统中的通用服务（系统级服务）：
  - 命名服务、目录服务、持久服务、安全服务、事务服务、持久性服务……

例：CORBA的体系结构



面向对象的分布式调用相关概念

- **接口（Interface）：**

  接口是系统中用来定义分布式对象能力的约定

- **对象引用（Object Reference）**

  对象引用是在客户程序的编程语言或脚本环境内用来引用分布式对象的实例。对象引用不同于OO中的对象指针概念

- **对象创建（Object Creation）**

  分布式对象系统必须为创建一个新的分布式对象实例提供一种机制。工厂（factory）是一种特殊的分布式对象类型，常用来创建其它的分布式对象

面向对象的分布式调用相关概念(续)

♦ 对象调用（Object Invocation）

分布式对象系统必须为分布式对象的调用提供一种机制

♦ 对象撤销（Object Destruction）

分布式对象系统必须提供一种机制，一旦一个分布式对象的实例不再被使用，就必须及时的将它从系统中删除。

- COM提供了分布式引用计数和垃圾回收方法
- CORBA没有对此提供系统级支持
- EJB明确定义了各种Bean的生命周期管理策略
- Web Service没有对此提供系统级支持

分布式系统中的系统级服务

- 除了通信协议外，分布式系统还会用到一些可能很复杂，但又会经常重复使用的服务

  如安全性管理、事务处理等。

- 一个平台或体系结构所显式提供的服务越多，开发者就越容易在更短的时间内开发出高质量的分布式系统
- 有了平台提供的服务，开发者可以将更多的精力集中于系统的商业逻辑（如**EJB**）
- 可通过交易功能提供通用服务

  e.g. 事务处理服务、时钟服务、管理服务、定时服务、安全服务，**email** 服务 **…**

常见的服务(1)

- **命名服务（Naming）**

  在分布式系统中，命名服务提供了一种定位分布式对象的机制。

- **监视（Monitoring）**

  监视服务不仅可以监视系统的运行状态，而且当需要操作人员参与时可以发出警告信息。

- **许可（Licensing）**

  许可服务用于确认分布式系统的用户已经购买了适当的使用许可。

- **持久性（Persistence）**

  持久性服务提供一种统一的机制，使得分布式对象可以通过持久的数据存储来保存、更新和恢复它的状态

常见的服务(2)

- **安全性（Security）**

  安全性服务确保于分布式对象的通信是安全的，并确认相应的用户具有适当的权限。

- **事务（Transaction）**

  事务服务能够确保一个事务或者完全完成，或者完全放弃。在企业系统中，事务定义了工作的原子级（atomic）单元。分布式事务处理就是一个跨越多台计算机的单个工作单元。

- **消息处理（Messaging）**

  消息处理服务提供异步编程模式。异步模式在很多应用中都需要。

常见的服务(3)

- **分布式垃圾回收（Distributed garbage collection）**

  当一个程序不再使用分布式对象时，分布式垃圾回收服务会自动释放分布式对象所占用的存储单元。

- **资源管理（Resource Management）**

  一般来说，资源管理器按照使可伸缩性最大化的方式来管理分布式对象，即支持大量的客户程序同分布式对象在短时间内进行交互的能力。

分布对象技术三大流派

- OMG 的 CORBA：Common Object Request Broker Architecture
- 微软的COM / DCOM：(Distributed) Component Object Model

  (在Internet 上， .NET 框架， 基于XML 的Web Service)
- SUN的 EJB / J2EE