

5

分布式计算结构

5 分布式计算结构

- 分布式体系结构
- 中间件
- CORBA体系结构简介

5 分布式计算结构

- 分布式体系结构

支持硬件系统的计算平台由多个独立的计算元素组成。

- 多独立节点
- 资源可能无法直接访问
- 软件并发
- 多路控制/多路故障

5 分布式计算结构

- 分布式系统的优点

资源共享 (Resource sharing)

共享硬件和软件资源。

开放性 (Openness)

可使用有不同供应商提供的设备和软件。

并发性 (Concurrency)

可通过并发处理来增强性能。

缩放性 (Scalability)

可通过增加新的资源来提高生产力。

容错 (Fault tolerance)

在出现某个失误之后仍具备继续运行的能力。

5 分布式计算结构

- 分布式系统的缺点

复杂性 (Complexity)

分布式系统比集中式系统复杂得多。

保密性 (Security)

更易受到外部攻击。

可管理性 (Manageability)

需要更多的人力来管理系统。

不可预见性 (Unpredictability)

响应结果难于预料，与系统构造和网络的负载情况有关。

5 分布式计算结构

- 定义

分布式体系结构是一种支持应用程序开发的体系结构，它可以利用由多个独立的处理单元组成的物理体系结构。这些单元不共享内存，而是通过网络发送消息相互协作。

关键特征：

- 多个独立组件
- 组件不是由所有用户共享的
- 资源可能无法访问
- 软件运行在不同处理器上的并发进程中
- 多个控制点
- 多个故障点

5 分布式计算结构

中间件系统

中间件是一种支持交互通信的软件。它提供了一个API，将应用程序代码与底层网络通信格式和协议隔离开来。

中间件系统通过在网络中创建统一和一致的错觉来实现各种形式的分布式。换句话说，所谓的“single-system image”。

它们充当独立组件和过程之间的粘合剂。（例如客户端，服务器），通过在多个操作系统之上提供普适服务。

5 分布式计算结构

三种中间件系统：面向事务的中间件、面向消息的中间件和面向对象的中间件

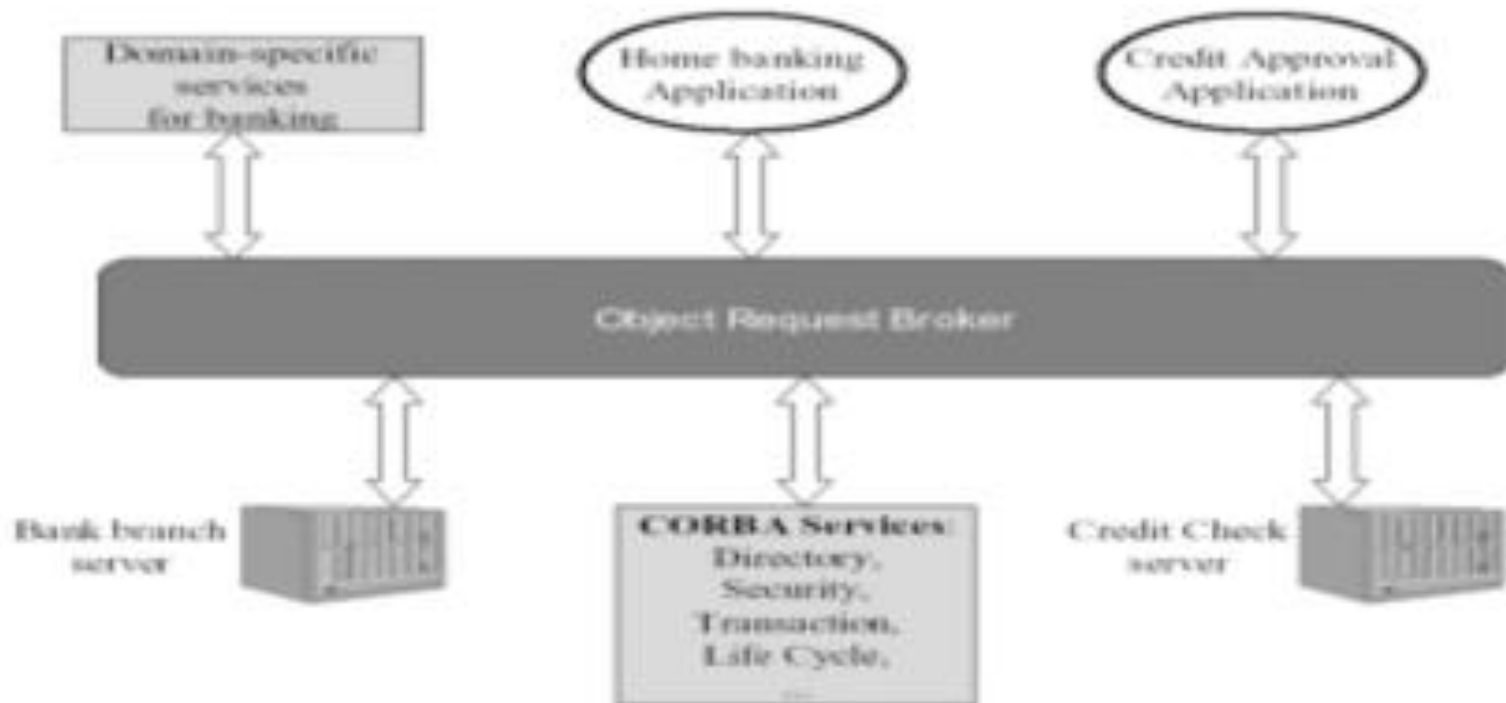
- 面向事务的中间件支持涉及数据库应用程序的并行计算。
- 面向消息中间件支持分布式组件之间可靠的异步通信。
- 面向对象的中间件系统是基于面向对象范式的，并且主要支持组件间的同步通信。

目前最流行的面向对象中间件包括CORBA、CIM、和EJB（基于CORBA的中间件系统），也称为面向对象的中间件。

- 基于远程过程过程调用（RPC）基础
- 扩展了RPC框架，引入了面向对象的机制。

5 分布式计算结构

例：基于CORBA的分布式中间件



5 分布式计算结构

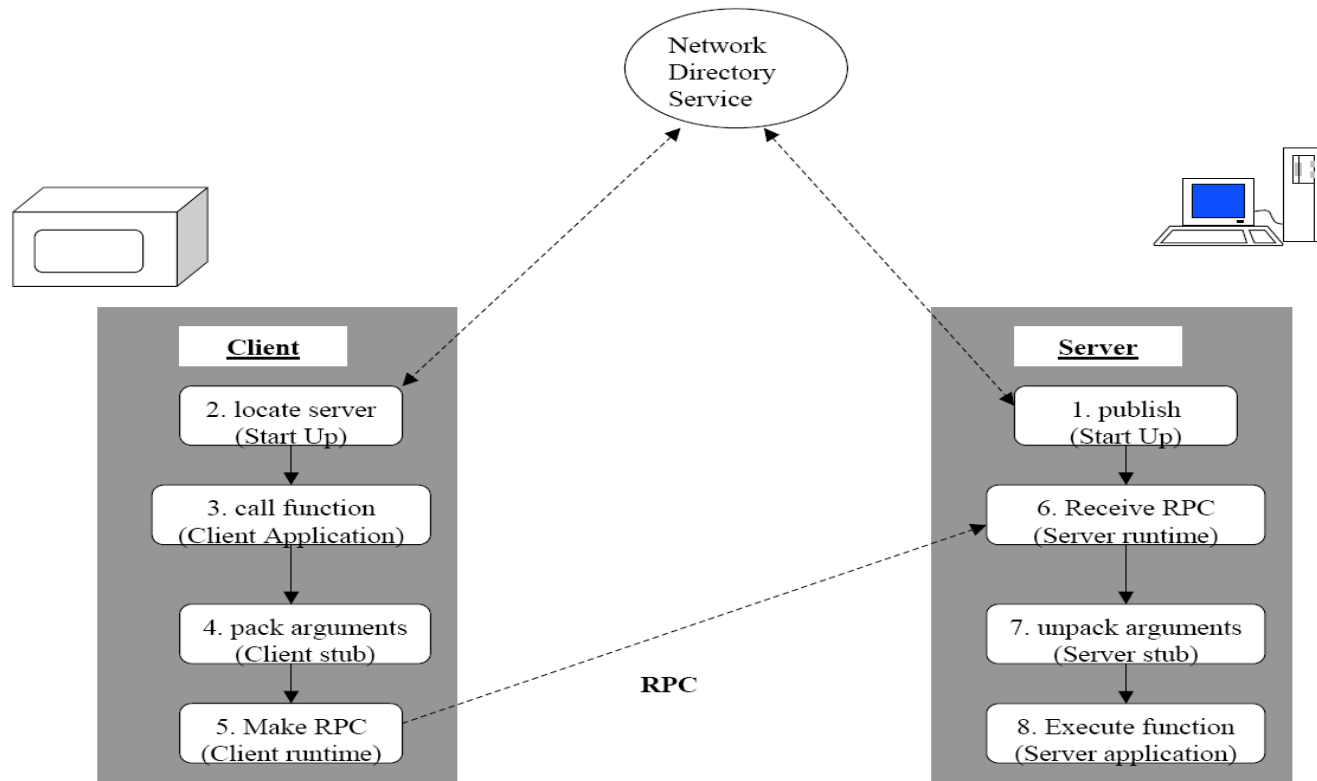
远程过程调用(RPC)

- RPC允许跨不同的硬件和操作系统平台调用操作。
- 提供与本地过程调用相同的机制，但在进程间级别。
- 支持通用接口定义语言(IDL)。

RPC机制由RPC软件提供，它“透明地处理所有涉及的步骤”。RPC软件的功能包括：服务器函数和并发请求的位置、参数传递和数据表示、故障管理、安全管理。

5 分布式计算结构

RPC Mechanism



5 分布式计算结构

CORBA—通用对象请求代理架构

CORBA是一个对象请求代理的一种国际标准，它是管理分布式对象通信的中间件。

对分布计算中间件的需求处在两个级别上：在逻辑通信级别上，中间件允许对象在不同的计算机上交换数据和控制信息；在构件级别上，中间件为开发兼容构件提供一个基础，CORBA构件标准已经定义好。

5 分布式计算结构

CORBA技术标准

- 一个针对应用对象的对象模型：一个CORBA对象是对状态的一个封装，它具有定义好的接口（用接口定义语言IDL），并且这种接口与语言类型无关。
- 由一个对象请求代理来管理对象服务请求。
- 有一套常规对象服务供许多分布式应用使用，还有建立在这些服务之上的一套公共构件。

5 分布式计算结构

CORBA对象 (Object)

- 原则上，CORBA对象可以与C++和Java对象进行比较。
- 它们必须有一个单独的接口定义，该定义采用与C++类似的通用语言(接口定义语言-IDL)。
- 这种接口定义语言 (IDL) 可以映射到编程语言上 (C++, Java等)。因此，用不同语言编写的对象可以相互通信。

5 分布式计算结构

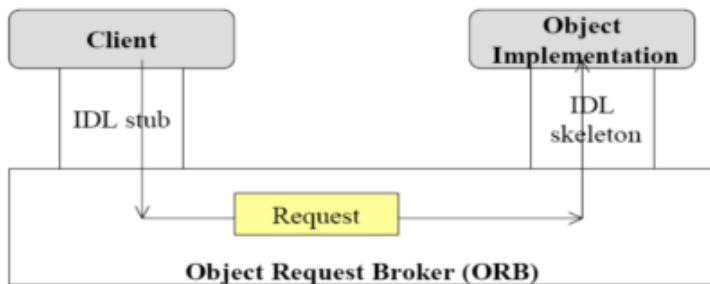
CORBA对象请求代理 (ORB)

- 由对象请求代理 (ORB) 操纵对象通信。它了解全部在系统中的对象及其接口。
- 在使用一个ORB的时候, 呼叫对象附有一个IDL 票根 (stub), 上面定义了被叫对象的接口。
- 调用这个票根也就调用了ORB, 随后ORB 发布一个IDL 主票 (skeleton) 来调用所需对象, 由此连接到实现该服务的接口。

5 分布式计算结构

CORBA体系结构

- 一个object所提供的服务是在一个充当它与系统其余部分之间的接口的契约中表达的。
- 对象接口使用一种名为接口分解语言(interface definition language, IDL)的特殊语言表示。
- 要让对象通过网络进行通信，它们需要一个名为Object Request Broker (ORB)的通信基础设施。
- 客户端和对象实现都通过IDL接口与ORB隔离。客户端只查看对象的接口，而不查看实现。
- 通信请求不会直接从客户端传递到对象实现;相反，每个请求都被传递到客户机的本地ORB, ORB负责管理它。

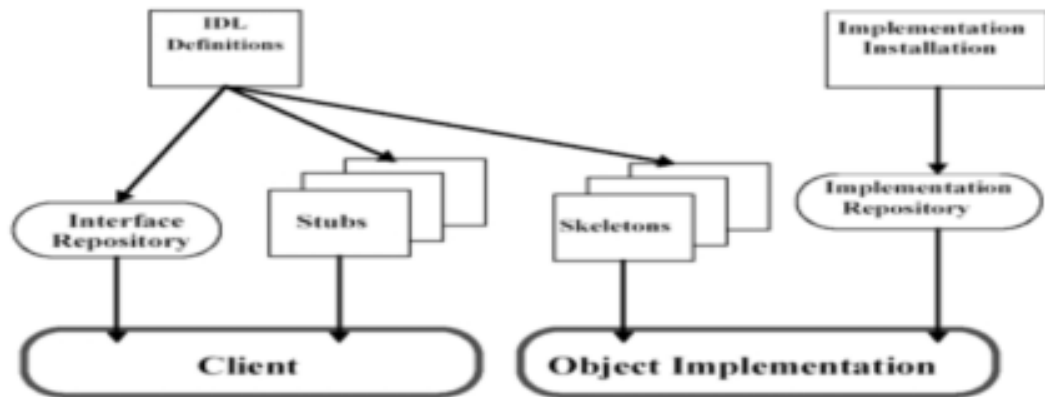


5 分布式计算结构

接口定义语言(IDL)

IDL提供了一种与编程语言无关的方法来定义服务是如何实现的。是规范语言之间的一种中介语言。如UML和C、C++ 等编程语言。

- 提供了客户机将使用和服务器将实现的接口的抽象表示。
- 客户端和对象实现由三种机制隔离:客户端上的IDL存根、ORB和实现端上的相应框架。



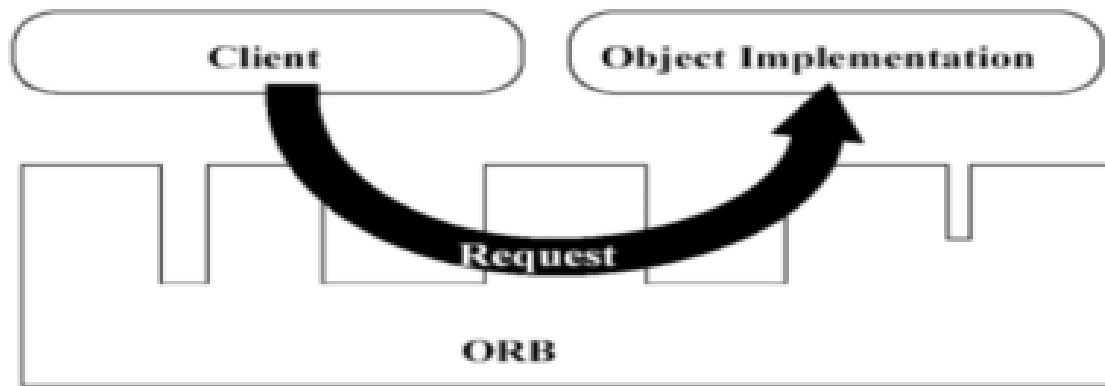
5 分布式计算结构

对象请求代理

客户端看到的接口完全独立于对象所在的位置，它是用什么编程语言实现的，或者没有反映在对象接口中的任何其他方面。

ORB负责：

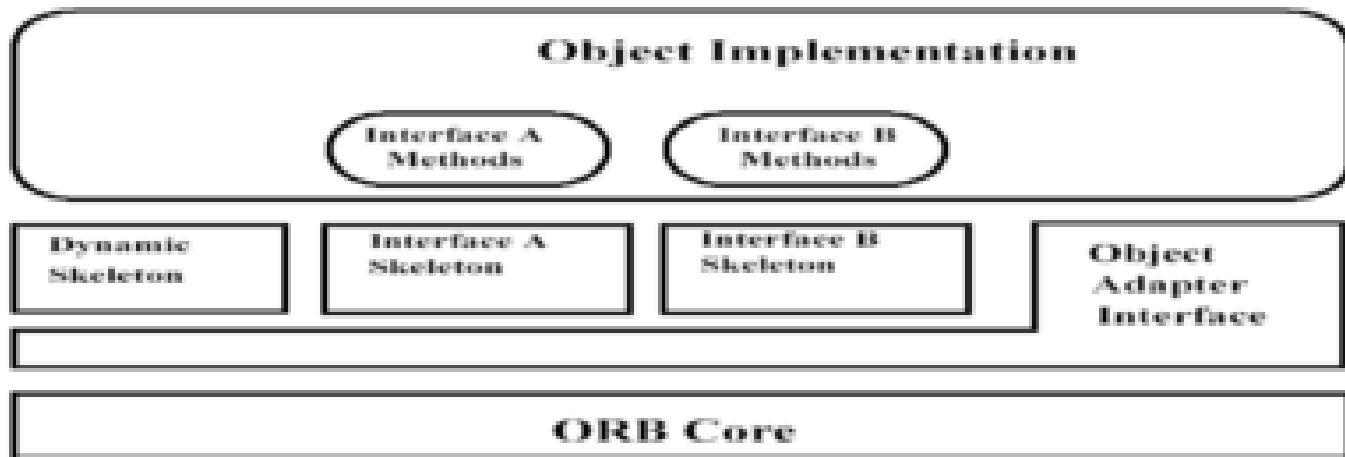
查找请求的对象实现，准备对象实现来接收请求，传递组成请求的数据。



5 分布式计算结构

对象适配器

- 对象适配器是对象实现访问ORB服务的主要方法；
- 对象适配器负责以下功能: 对象引用的生成和解释、方法调用、安全的交互、对象和实现激活和取消激活、将对象引用映射到相应的对象实现、注册的实现。



5 分布式计算结构

公共对象服务(COS)

目前最多有15个服务可以帮助ORB，它们定义在ORB之上，作为具有“IDL接口”的标准CORBA对象。

网站服务内容包括：

- 命名服务: 为CORBA客户机(和服务服务器)提供了一种查找网络上的对象的方法。
- 事件服务: 存储和交付客户端或服务服务器发送的事件。
- 安全服务: 提供对消息进行身份验证、授权对象的访问和提供安全通信的方法。
- 事务服务: 定义控制针对数据库或其他子系统的操作方法。

5 分布式计算结构

CORBA命名服务

面向对象的中间件使用对象引用来处理服务器对象。我们需要找到一种方法，在不假设物理位置的情况下获得这些对象引用。

Name是可以绑定到对象引用的字符串序列，
对象引用-可以解析名称绑定以获取对象引用。
CORBA命名服务: 允许通过外部名称定位组件;
CORBA交易服务: 根据服务特征定位组件。