北京郵電大學

分布计算环境

北京邮电大学计算机学院

Chapter 1

Chapter 1

Introduction

主要内容

- 分布式系统及其挑战
- 什么是分布式计算？
- 什么是分布计算环境
- 分布计算环境的发展历程
- 课程简介

什么是分布式系统？

- A collection of independent computers that appears to its users as a single coherent system.
- A collection of autonomous computers linked by a network, with software designed to produce an integrated computing facility.
- the system software runs on a loosely integrated group of cooperating processors linked by a network.
- 在网络计算平台上开发、部署、管理和维护以资源共享和协同工作为主要应用目标的分布式应用系统。
- ……

分布式系统举例

- 分布式系统到处可见
  - 校园的图书管理系统、行政办公系统、ATM系统、Internet、Web、PSTN、3G、4G、5G ……
  - 甚至大作业、毕设参与开发的系统
- 不同的系统会在多个方面有差别，如
  - 从规模上：LAN、WAN
  - 从行业上：银行网络（ATM） 电信网络（IP Phone）
  - 从边界上：Internet、Intranet
  - 从协议上：TCP/IP
  - ……
- 在IT领域，目前特别关注以IP为核心实现的网络环境、互联网环境、Web环境、移动互联网环境、物联网环境等等

分布式系统的主要特征

- 包含任意个数的系统进程和用户进程
- 体系结构模块化，它由数目可变的多个处理部件组成
- 通过共享通信结构上的报文传递进行通信，进程之间的报文传送存在延迟且延迟时间可变
- 实行某种全系统范围的控制，以便提供动态的进程间的合作和运行时间的管理
  - 不同系统有着不同的控制程度

分布式系统的挑战

- 一般，分布式系统需要支持以下特性
  - 资源共享
  - 开放性
  - 可伸缩性
  - 并发性
  - 容错性
  - 透明性

资源共享

- 一旦授权，可以访问系统中的任何资源：
  - 硬件(e.g. printer, scanner, camera，computer)、软件（服务）、数据(file, database, web page)
- 相关技术例：
  - 资源管理器控制资源的访问
    - ➔ 提供命名机制
    - ➔ 控制并发访问

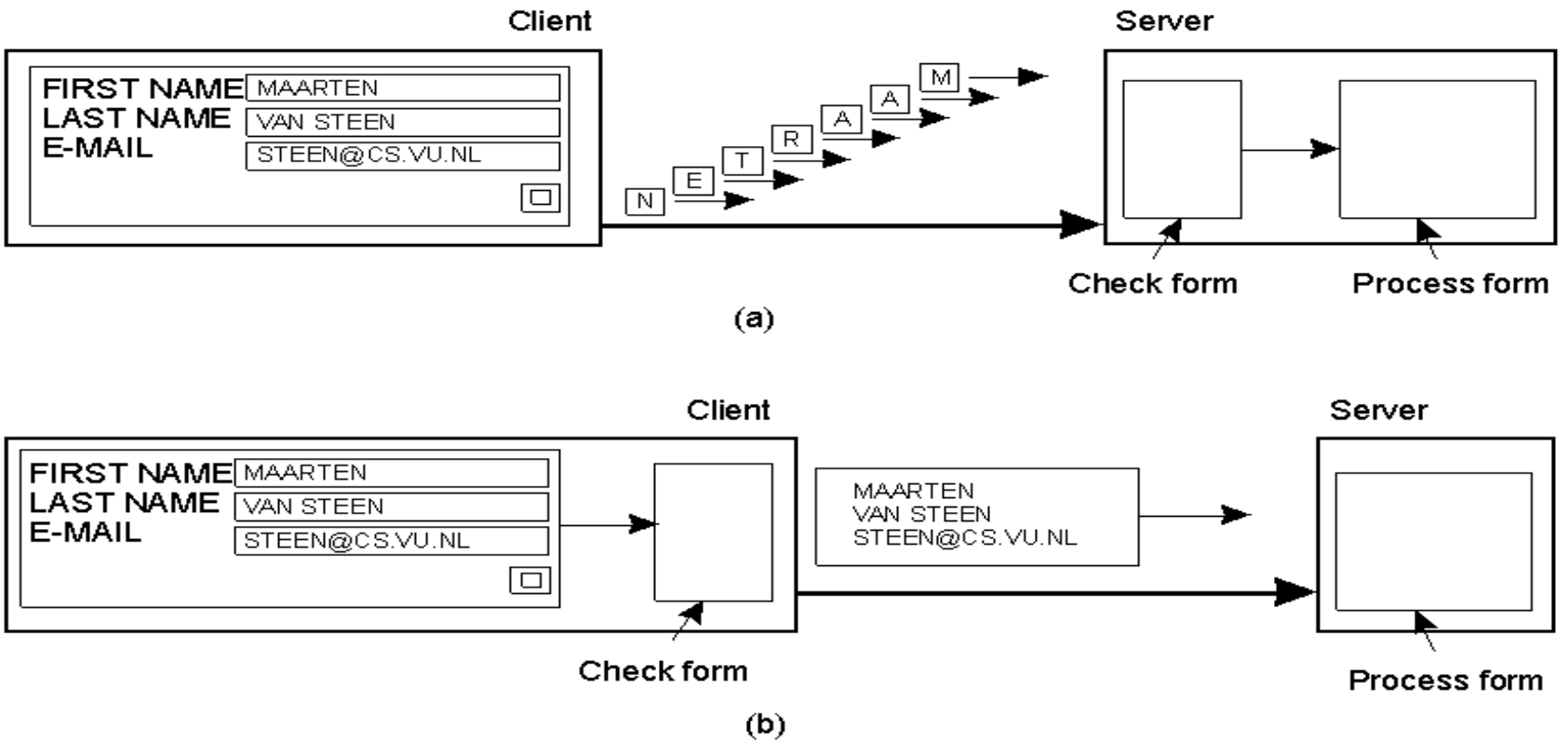
开放性

- 新共享资源添加并被各种客户程序使用的（难易）程度
  - 支持异构资源的添加和使用
- 相关技术例
  - 发布系统的关键接口
  - 提供统一的通信机制
  - 发布访问共享资源的接口
  - 虚拟化技术

可伸缩性

- 在资源和用户数较大增长的情况下，系统性能仍能维持原状
  - 利用网络环境可以为更多的用户服务、而且响应更快
  - 通常通过增加更多/更快的处理器、更可靠、更完善的服务实现
- 当环境增长时，系统组成部分不必改变，所设计的各个组成部分应当便于系统的伸缩
  - 模块化
    - 功能
    - 数据

Scaling Techniques (1)：功能分布

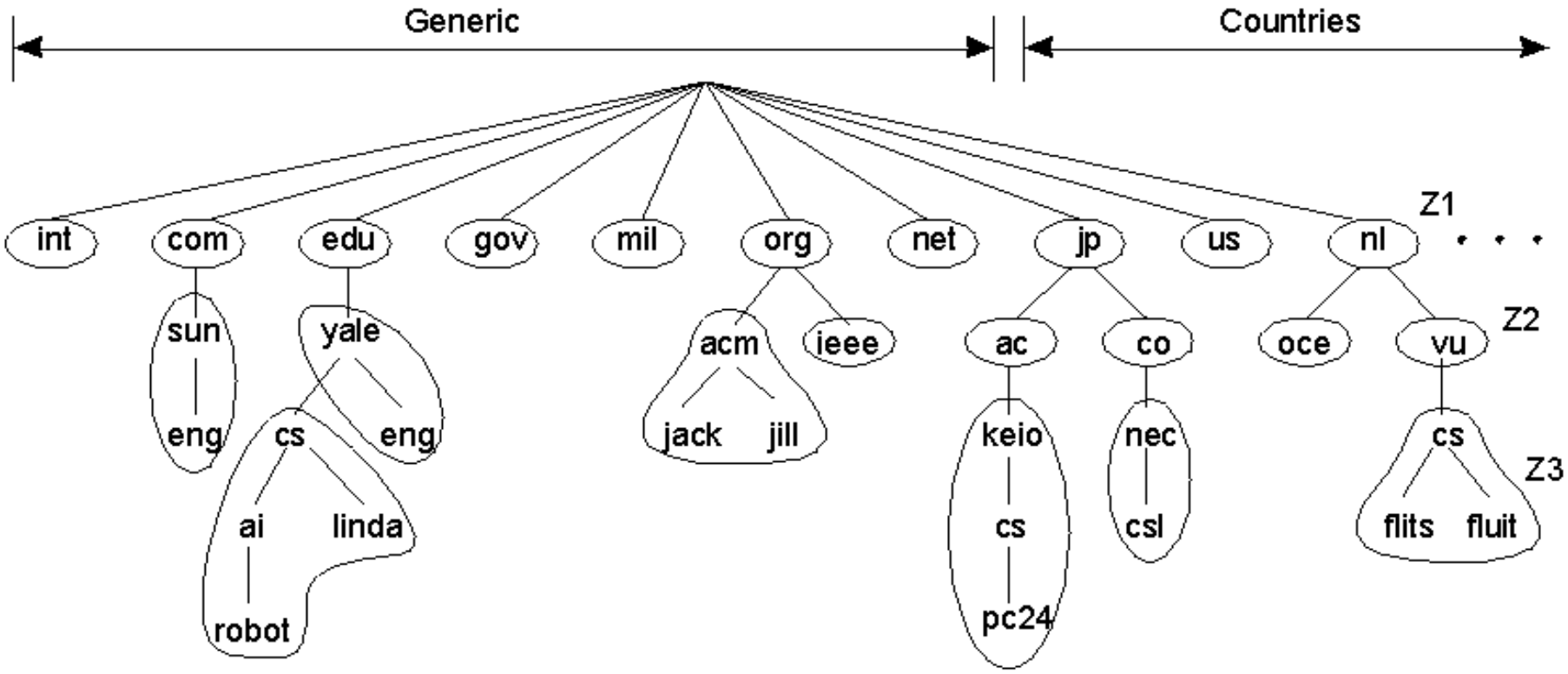


The difference between letting:

a) a server or

b) a client check forms as they are being filled

Scaling Techniques (2)：数据分布



An example of dividing the DNS name space into zones.

并发性

- 分布系统中的各个组成部分可以在并发的过程中被执行，如：
  - 多个用户同时访问应用程序
  - 多个服务进程同时运行，相互协作
- 分布系统中的各个进程可以并发访问、更新共享的资源
- 相关技术例
  - 并发控制

容错性

- 错误发生时，系统能够继续工作的能力
  - 硬件、软件、网络发生错误的不可避免性
- 相关技术
  - 恢复 Recovery
    - ➔如：数据库的Roll back
  - 冗余 Redundancy
    - ➔如： IP route, replicated name table of DNS

透明性（1）

- 分布式系统对于用户和应用程序而言，应该是一个整体，而不是一个互相协作的简单的构件集合
  - Sun：Network is computer
- 透明性具有多重的内容，例如
  - 访问透明性　位置透明性　重置透明性　移动透明性
    持久透明性　复制透明性　故障透明性 事务处理透明性 。。。
- 相关技术
  - 繁多

透明性（2）

- 访问透明性：用相同的操作访问本地资源和远程资源
  - NFS中的文件系统操作（ a graphical user interface with folders ）
  - Web环境中的航行
  - SQL查询
- 位置透明性：不需要知道资源的物理位置就可以访问
  - NFS中的文件系统操作（ a graphical user interface with folders ）
  - Web中的页面（URL）
  - 分布式数据库中的表
  - Email系统

透明性（3）

- 重置透明性：在向资源发起请求时，可以屏蔽掉资源是否位置发生改变
  - 遇忙呼叫转移
- 移动透明性：资源或客户能够在系统中移动，而不会影响用户或程序的运行
  - 移动终端
- 持久透明性：客户使用资源时，不用关心资源的激活或者去激活过程
  - OS的页面管理

透明性（4）

- 复制透明性：使用资源的多个副本提升性能和可靠性，而客户无需知道副本的相关信息
  - 分布式DBMS
  - Web页面镜像、Web-caching
- 故障透明性：对客户屏蔽资源的故障和故障恢复情况
  - 容错机
  - Email系统
- 事务处理透明性：客户只需对事务方案进行定义，而屏蔽掉复杂的事务处理机制
  - 数据库的事务支持功能

分布式系统的优势

- 连接不同的系统、不同的机构团体
- 通过互连和互操作提高系统的协作能力
- 通过并行处理、负载平衡等提高系统的性能
- 通过复制技术提高系统的可靠性和可用性
- 通过模块化技术提高系统的可伸缩性
- 通过动态配置和重新配置功能提高系统的可扩展性
- 通过资源共享提高系统的性能价格比
- ...

分布式系统的(潜在)问题

- 通信问题：

  信息丢失、恢复、网络过载 ...

- 安全问题：

  数据共享 vs 数据保密

- 质量问题：

  网站系统崩溃、性能达不到要求、云服务宕机.....

- 支撑软件需求：

  适合的分布计算环境及应用设计方法

小结

- **Distributed systems are everywhere**

  Internet, intranet, wireless networks.

- **Resource sharing is the main motivating factor for constructing distribute systems.**

- 分布式系统比想象的要复杂:
  - 异构环境下的应用互操作问题
  - 系统管理问题
  - 系统安全问题
  - 透明性支持问题
  - …….

主要内容

- 分布式系统及其挑战
- 什么是分布式计算？
- 什么是分布计算环境
- 分布计算环境的发展历程
- 课程简介

什么是分布式计算?

- 简单地说，分布式计算是两个或多个软件共享信息、协同工作（的过程）。这些软件既可以在同一台计算机上运行，也可在通过网络连起来的几台不同机器上运行。
  - 多个进程，通过网络技术进行通信
- 两种典型的应用途径
  - 将分布式软件系统看作直接反映了现实世界中的分布性
  - 用于改进某些应用程序的运行性能

分布计算技术是构造分布式系统的基础

并行计算和分布计算

- 并行计算（Parallel Computing）是指在并行的计算机上，将一个应用分解成多个子任务，分配给不同的处理器，各个处理器之间相互协同，并行地执行子任务，从而达到加快求解速度，或者提高求解应用问题规模的目的
- 并行计算突出的是时间上的同步性：同时进行计算
- 分布计算突出的是空间上的分布性：计算在不同的位置进行
  - 分布系统也可以用来做粗粒度的并行计算
- 两者之间有相同的支撑技术：通信、安全、容错……

主要内容

- 分布式系统及其挑战
- 什么是分布式计算？
- 什么是分布计算环境
- 分布计算环境的发展历程
- 课程简介

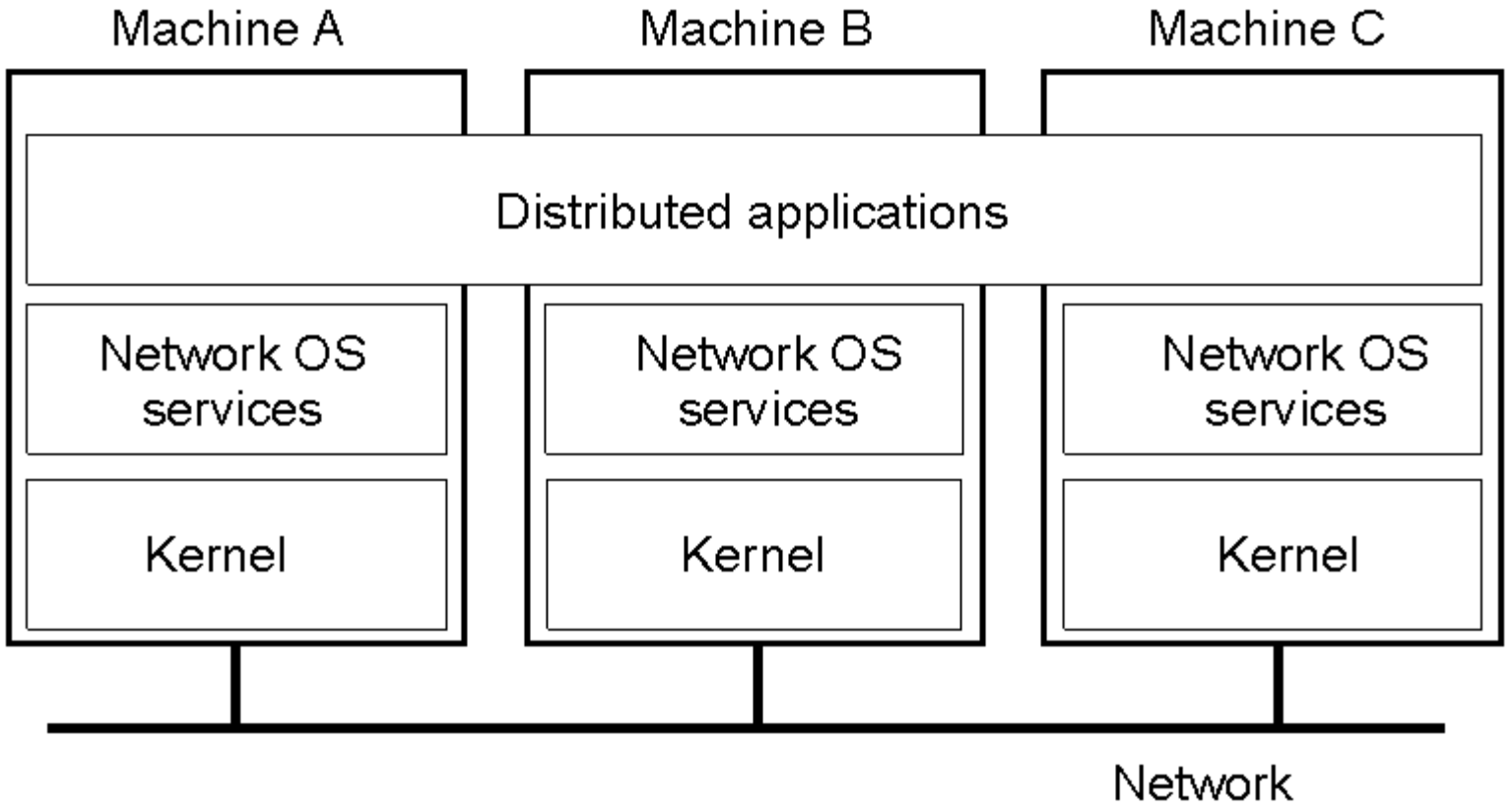
分布计算环境

- 分布计算环境提供了不同软、硬件平台资源共享和互操作的有效手段，使得分布式计算可以比较方便地得以实现，从而分布式应用系统可以比较方便地得以构造
  - 其自身常常就是一种分布式系统，可屏蔽硬件平台、操作系统、网络协议的差异性
  - 可方便分布式应用系统的构建：设计、实现、部署、维护
- 分布式计算环境构造的技术基础
  - 分布在网络上的程序之间的互操作技术
  - 目录技术、负载平衡技术、容错技术、事务管理技术、安全技术 ……

技术路线(1)

- **理想的技术路线**(上一世纪80年代学术界普遍追求的目标)

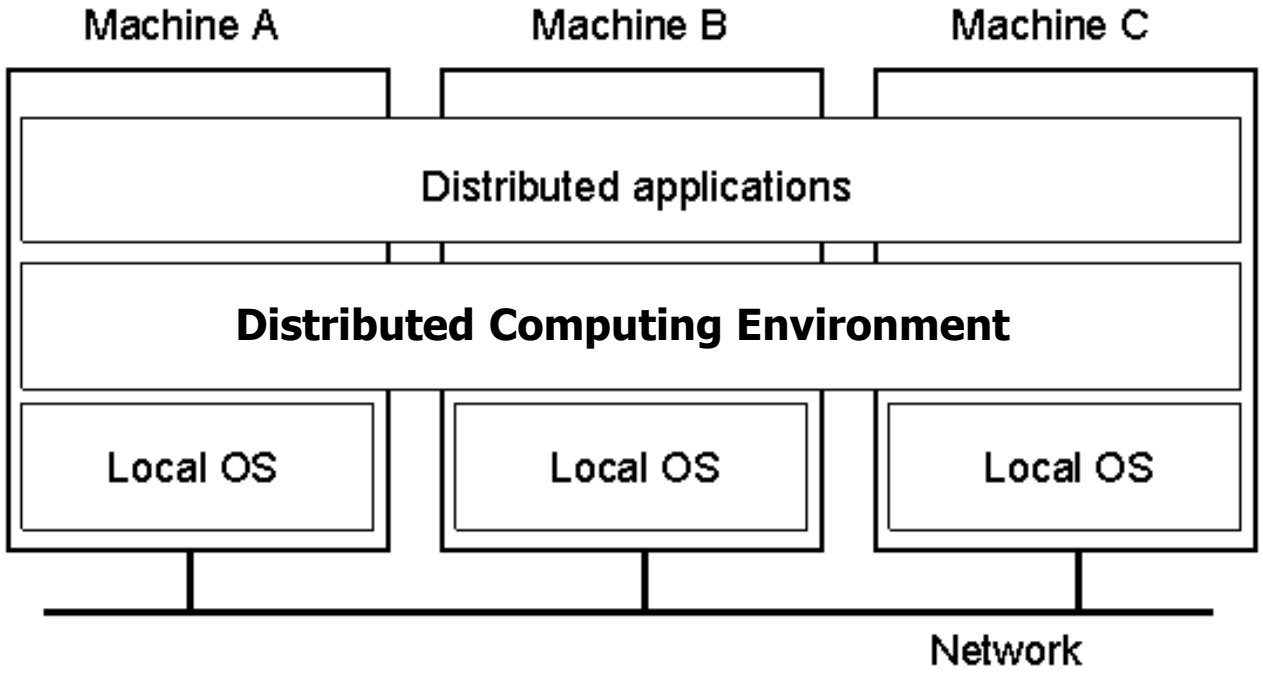
  试图在互连的计算机硬件上部署全新的分布式操作系统，全面管理系统中各自独立的计算机，呈现给用户单一的系统视图



技术路线(2)

现实的技术路线(上一世纪90年代之后业界普遍遵守的路线)

- 在网络计算平台上部署分布计算环境（中间件）
  - **屏蔽网络硬件平台、操作系统、网络协议等的差异性**
- 提供开发工具和公共服务
- 支持分布式应用
- 资源共享和协同工作



多类分布计算环境

- 面向不同的软件实现技术或目标应用场合，有不同的分布计算环境
  - 过程调用：DCE （RPC）
  - 面向对象（构件）：CORBA、DCOM、EJB、ICE、Spring
  - 面向消息通信：Kafka、RabbitMQ……
  - Web环境下：Web 1.0、Web 2.0、Web Service
  - 面向资源整合：P2P、网格计算、云计算、边缘计算
  - ……

主要内容

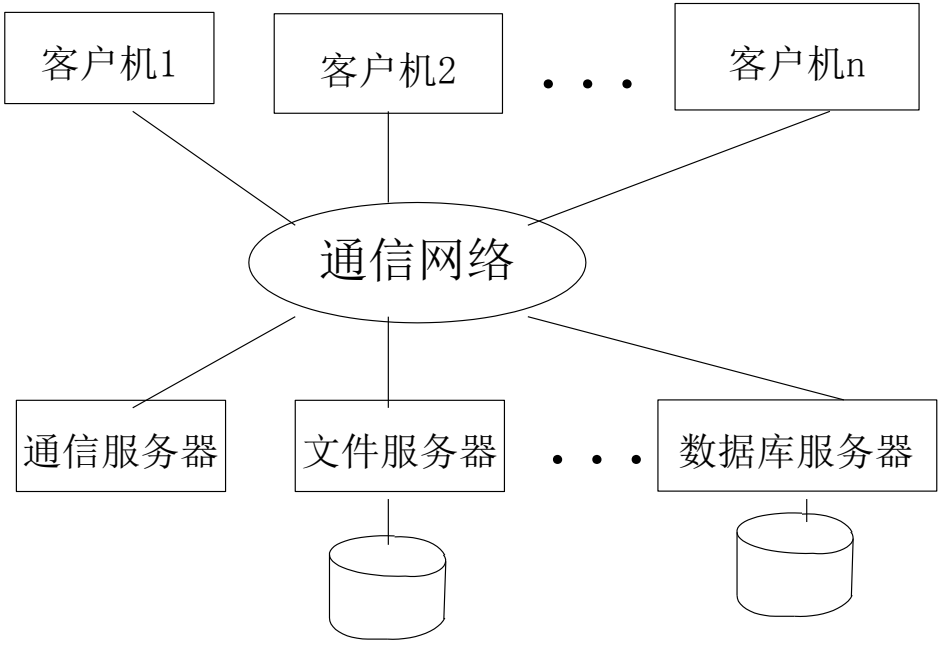
- 分布式系统及其挑战
- 什么是分布式计算？
- 什么是分布计算环境
- 分布计算环境的发展历程
- 课程简介

分布计算环境的发展历程

- 经典的C/S技术
- 远程过程调用RPC技术
- 面向对象的分布计算环境
- 面向构件的分布计算环境
- 面向Web的分布计算环境
- 面向资源整合的分布计算环境

Client/Server模式的基本形式

C/S模式是一种经典且基础的软件体系结构，把较复杂的计算和管理任务交给服务器，而把一些频繁与用户打交道的任务交给前端较简单的计算机—客户机。将任务合理分配到客户端和服务器端，既充分利用了两端硬件环境的优势，又实现了网络上信息资源的共享。强调客户端与服务端的分离

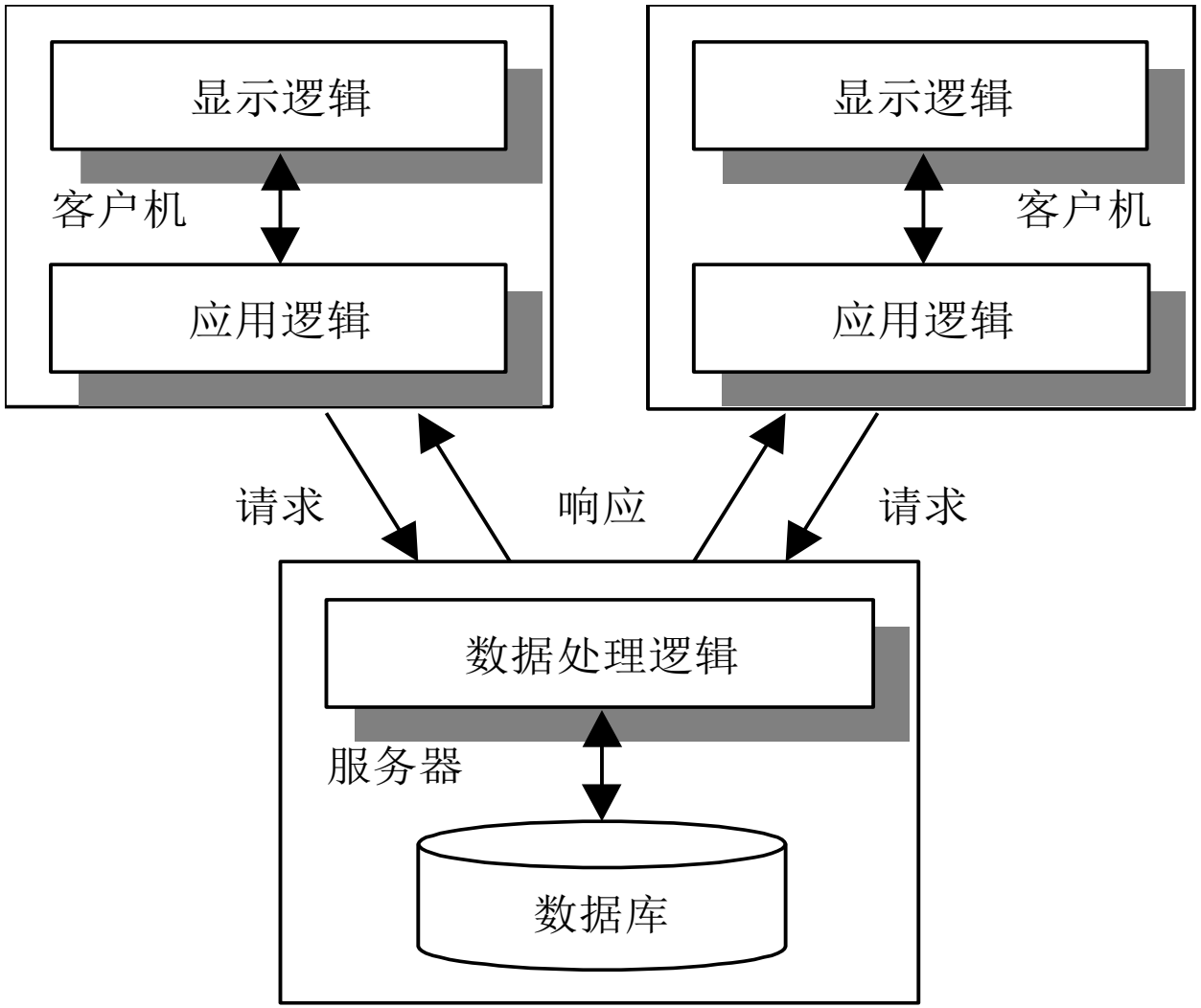


广义上讲，过程（函数、方法）的调用与实现也属于Client/Server关系

客户/服务器模式的层概念

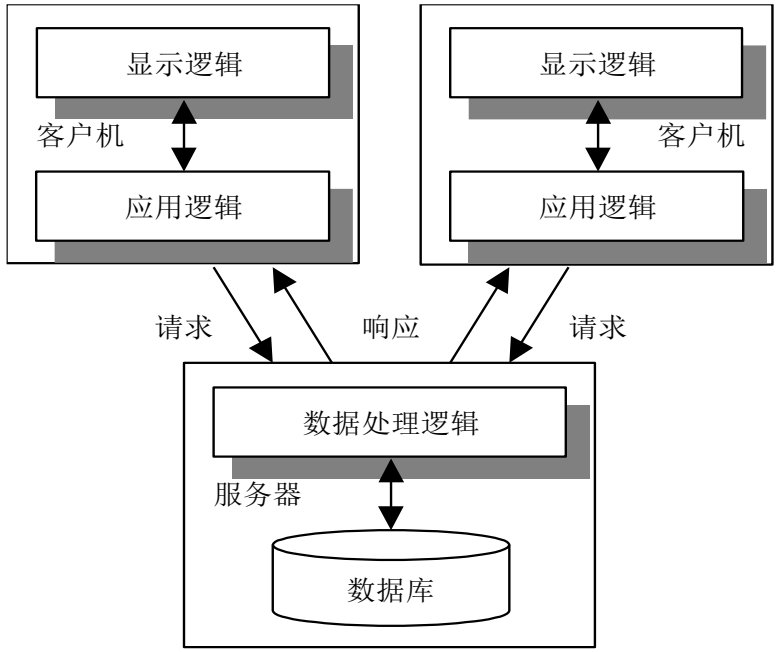
- 传统两层C/S模式
- 三层C/S模式
- N层 C/S 模式

传统两层C/S模式



两层模式的特点

- 特点
  - 请求应答方式
  - 以消息交换作为通信方式
  - 服务集中于特定Server
- 客户端与服务端分离带来的好处
  - 数据独立性
  - （客户端）平台无关性
    - ➔一定程度上的
  - 可扩展性、安全性、可靠性等



2层 ==〉 3层 ==〉 N层结构

- 2层结构存在很多缺陷：
  - 客户端的负担仍比较重（胖客户）

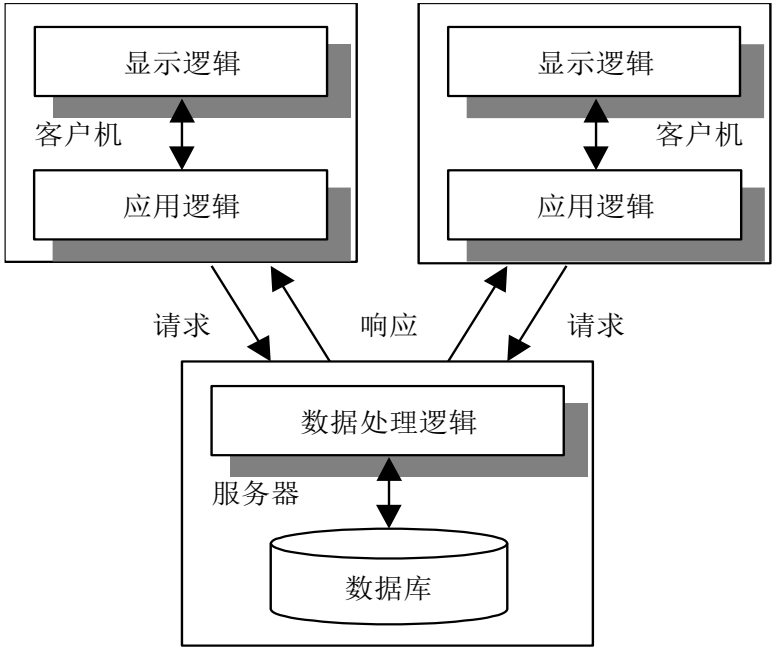
    仍然需要客户端进行较复杂的数据处理
  - 客户端的可移植性不好

    处理复杂必然牵涉更多的移植性问题

    每个客户端上都要安装数据库驱动程序
  - 系统的可维护性不好

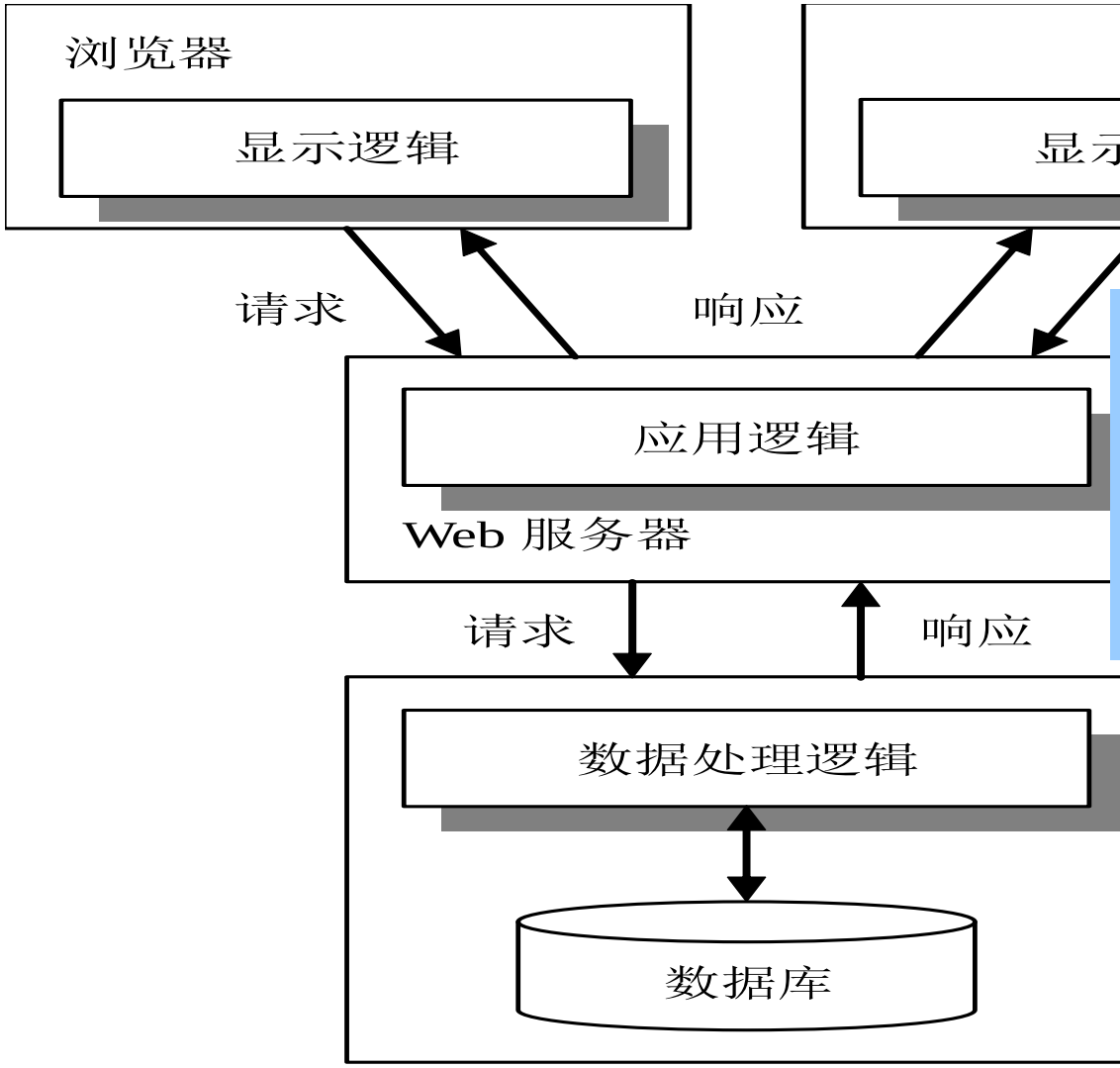
    客户端包含过多的商业逻辑

    商业逻辑与人机交互界面交织在一起
  - 数据的安全性



- 需求：需要更合理的工作分配——3层或多层结构

三层C/S结构

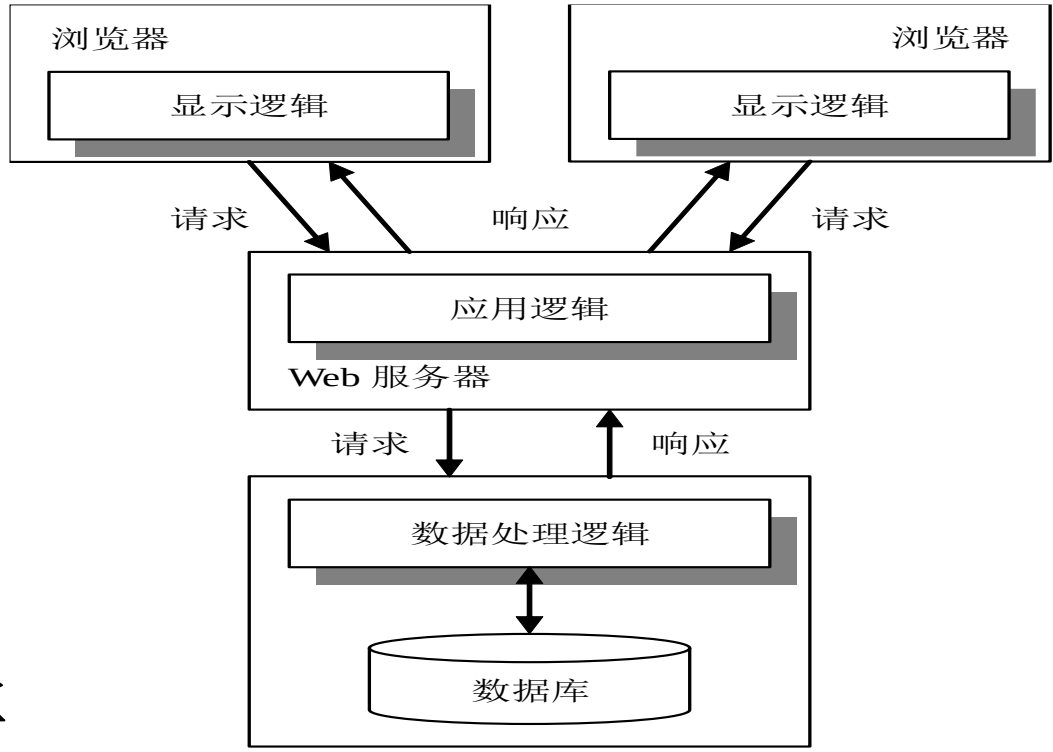


层的概念(TIERS)：在客户/服务器意义上，一个层就代表一个具有定义好的特定功能的平台，一个平台是一个计算机软件和硬件的组合

本图实际上是一个B/S 结构

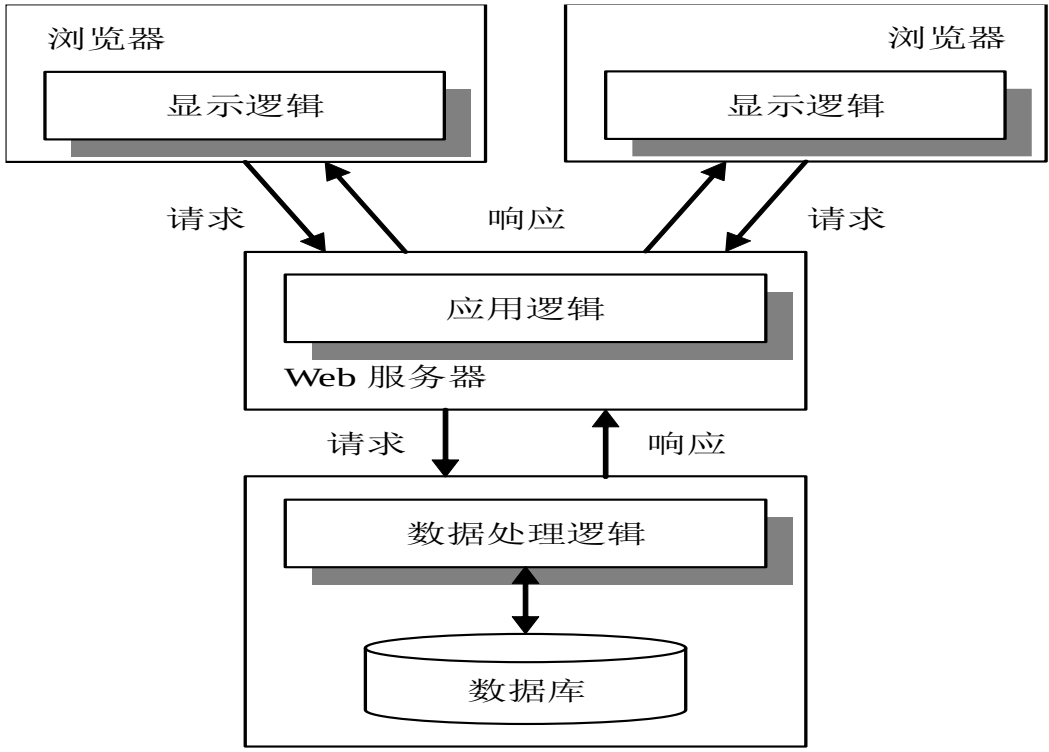
三层（多层）结构的主要优点

- 更合理的任务分配，层次清晰，管理和维护相对简单
- 使“胖客户”变成“瘦”客户，客户端只需把精力集中在人机界面上。
  - 前例的浏览器是纯粹意义上的“瘦”客户，也叫做B/S模式
- 中间业务逻辑层包含了大量的供客户端程序调用的业务逻辑规则(被用户共享)，可随具体业务的变化而改变，大大提高系统的可伸缩性



三层（多层）结构的主要优点(续)

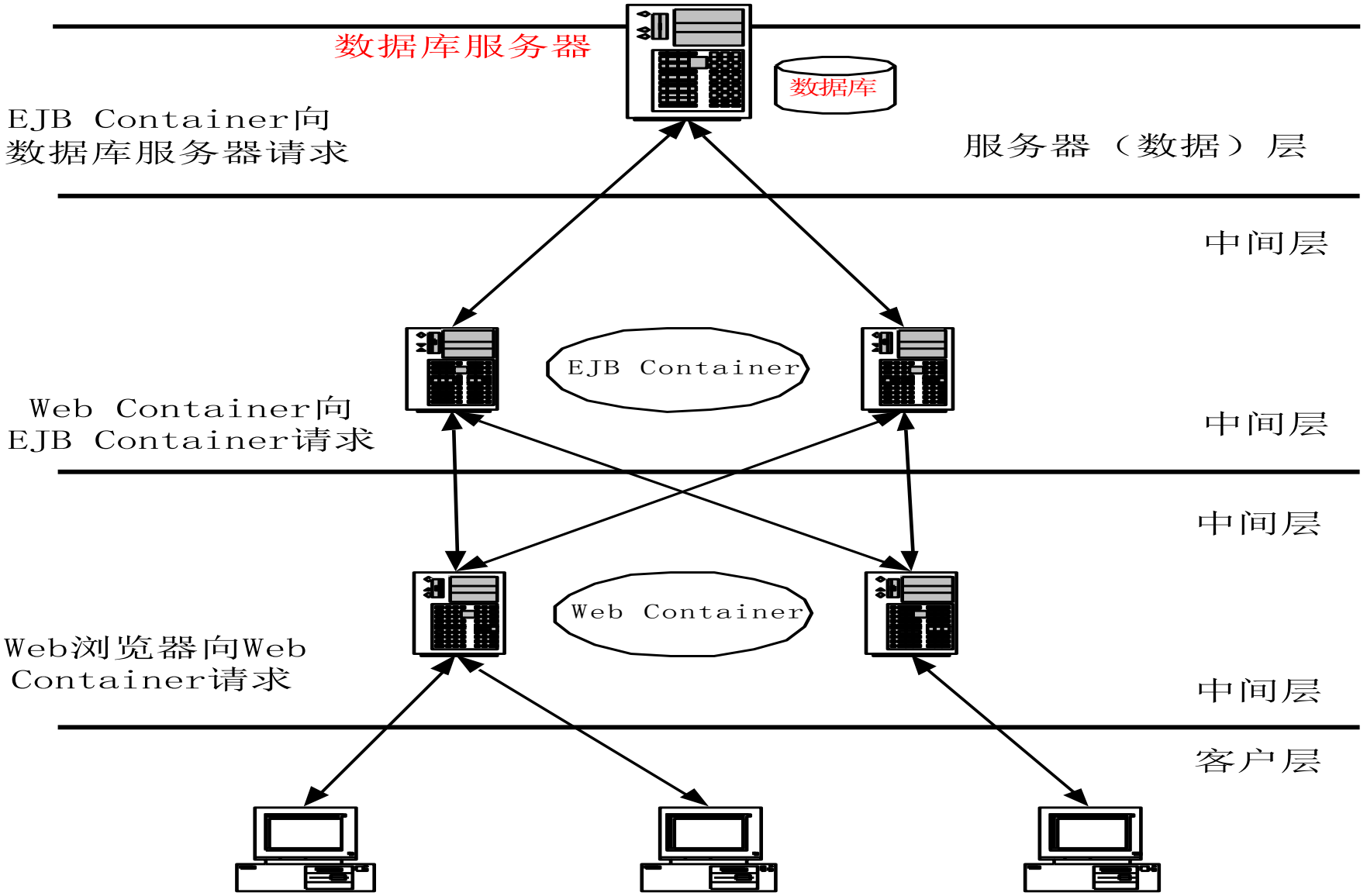
- 使中间层的业务逻辑处理与数据层的业务数据紧密结合在一起，可以提高系统的性能
- 数据服务层主要提供对数据库进行各种操作的方法，系统的安全性提高
- 大量的中间层分布计算环境提供丰富的系统级服务，使得开发人员可以以更少的工作量开发出更复杂、可靠、高效的软件系统



从三层到N层结构

- 在3层结构中，客户层和数据层已被严格定义，但中间层并未明确定义
- 中间层可以包括所有与应用程序的界面和持久数据存储无关的处理。假定将中间层划分成许多服务程序是符合逻辑的，那么将每一主要服务都视为独立的层，则3层结构就成为N层结构
- 例如，中间层可以分为实现界面呈现的Web服务器层和实现实际商业逻辑的EJB层

Java EE架构中的4层结构



层间通信

因为分布式软件跨越了多台计算机，所以需要一种类似于TCP/IP这样的网络基础设施来连接应用程序的各节点。

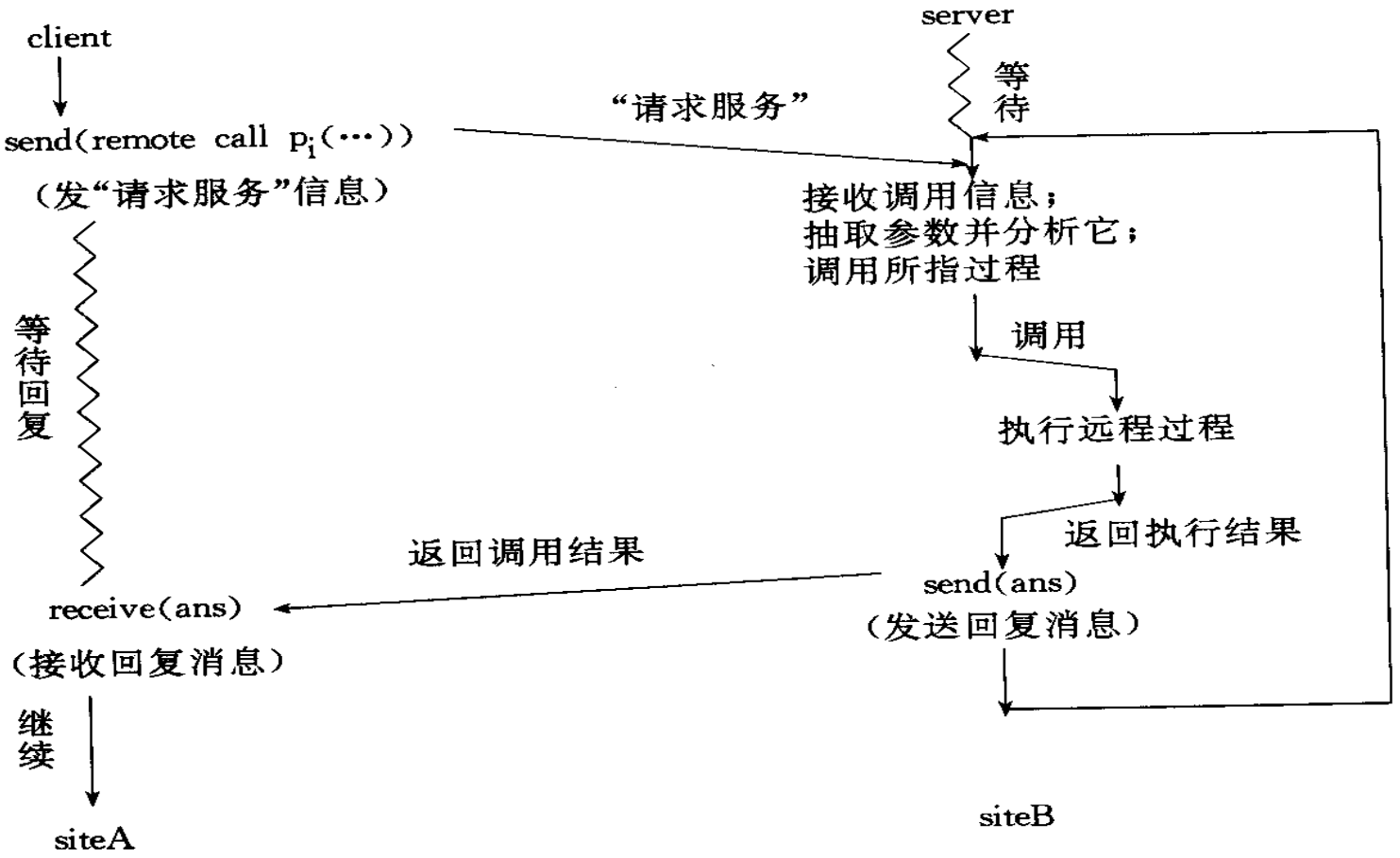
- 需要抽象层次更高、更方便实用的通信机制。
- 在两层结构中，对网络协议的抽象是比较容易的：
  - 数据库驱动程序
  - 嵌入式SQL
  - ODBC/JDBC等。
- 多层结构需要更复杂基础设施以实现跨网络的通信。
  - 相应的分布计算环境会进行支撑

分布计算环境的发展历程

- 经典的C/S技术
- 远程过程调用RPC技术
- 面向对象的分布计算环境
- 面向构件的分布计算环境
- 面向Web的分布计算环境
- 面向资源整合的分布计算环境

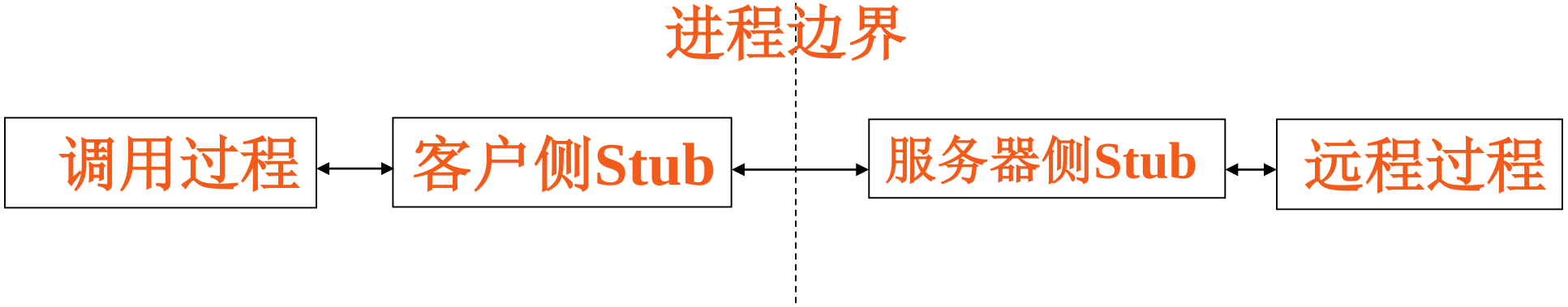
远程过程调用RPC

- 沿用用户熟悉的编程模式，调用远端过程并将结果返回。通信一般采用同步方式（Request-Wait-Reply）。



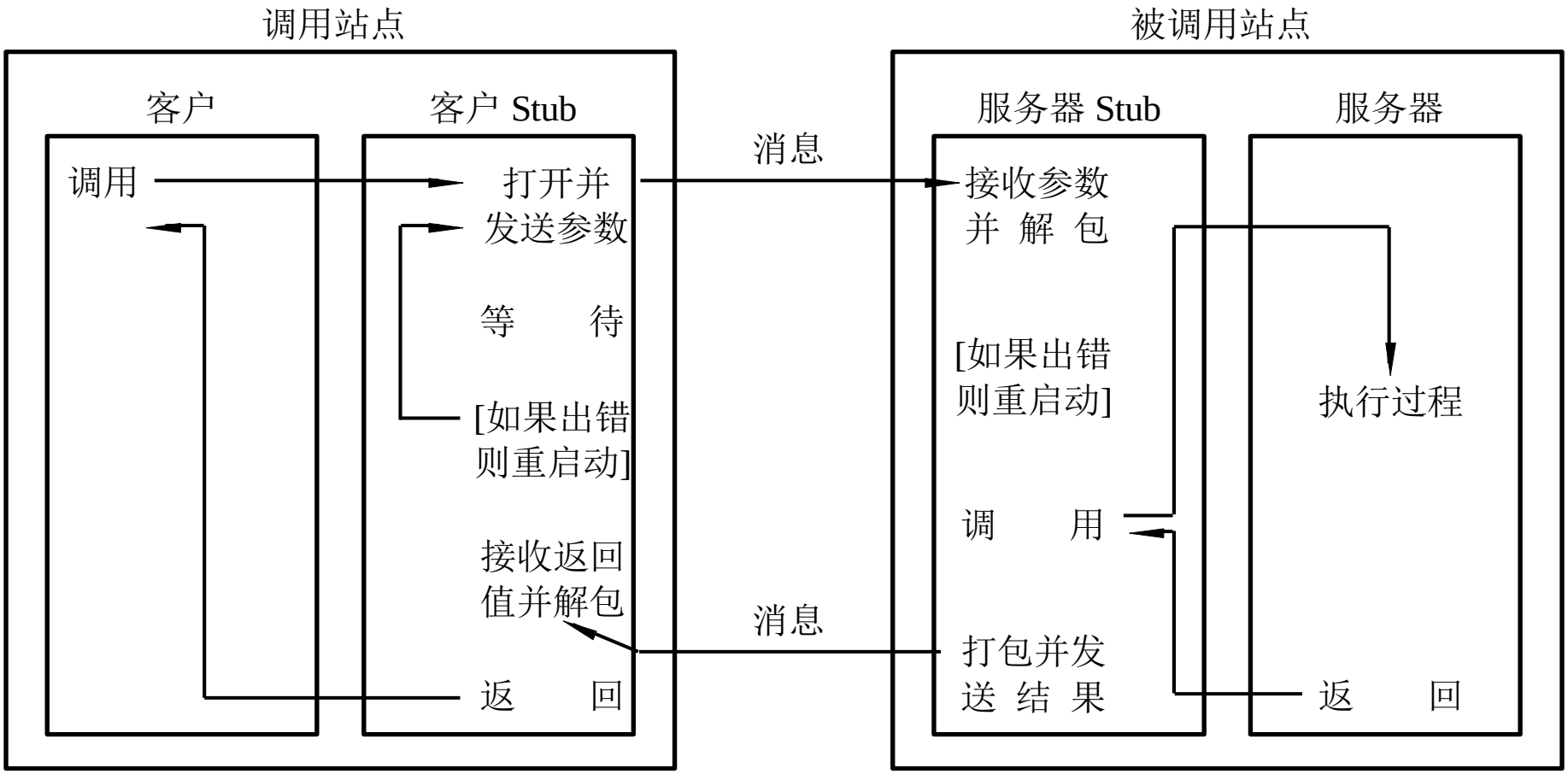
远程过程调用机制

- 在发送方（调用方）的地址空间里创建一个远程组件(如过程)的代理（客户侧Stub）；在接收方（被调用方）的地址空间里创建一个发送方代理（服务器侧Stub）。



- 发送方只和本地客户侧Stub通信；接收方从本地服务器侧Stub得到所有的请求。
- 组件间的远程通信封装在代理／Stub通信中，它是由建立在中间件API基础上的IDL编译器生成的。

远程过程调用机制(续1)


调用站点
被调用站点
客户
客户 Stub
服务器 Stub
服务器
调用
打开并
发送参数
等 待
[如果出错
则重启动]
接收返回
值并解包
返 回
消息
消息
接收参数
并 解 包
[如果出错
则重启动]
调 用
打包并发
送 结 果
执行过程
返 回

分布计算环境的发展历程

- 经典的C/S技术
- 远程过程调用技术
- 面向对象的分布计算环境
- 面向构件的分布计算环境
- 面向Web的分布计算环境
- 面向资源整合的分布计算环境

面向对象的分布计算技术

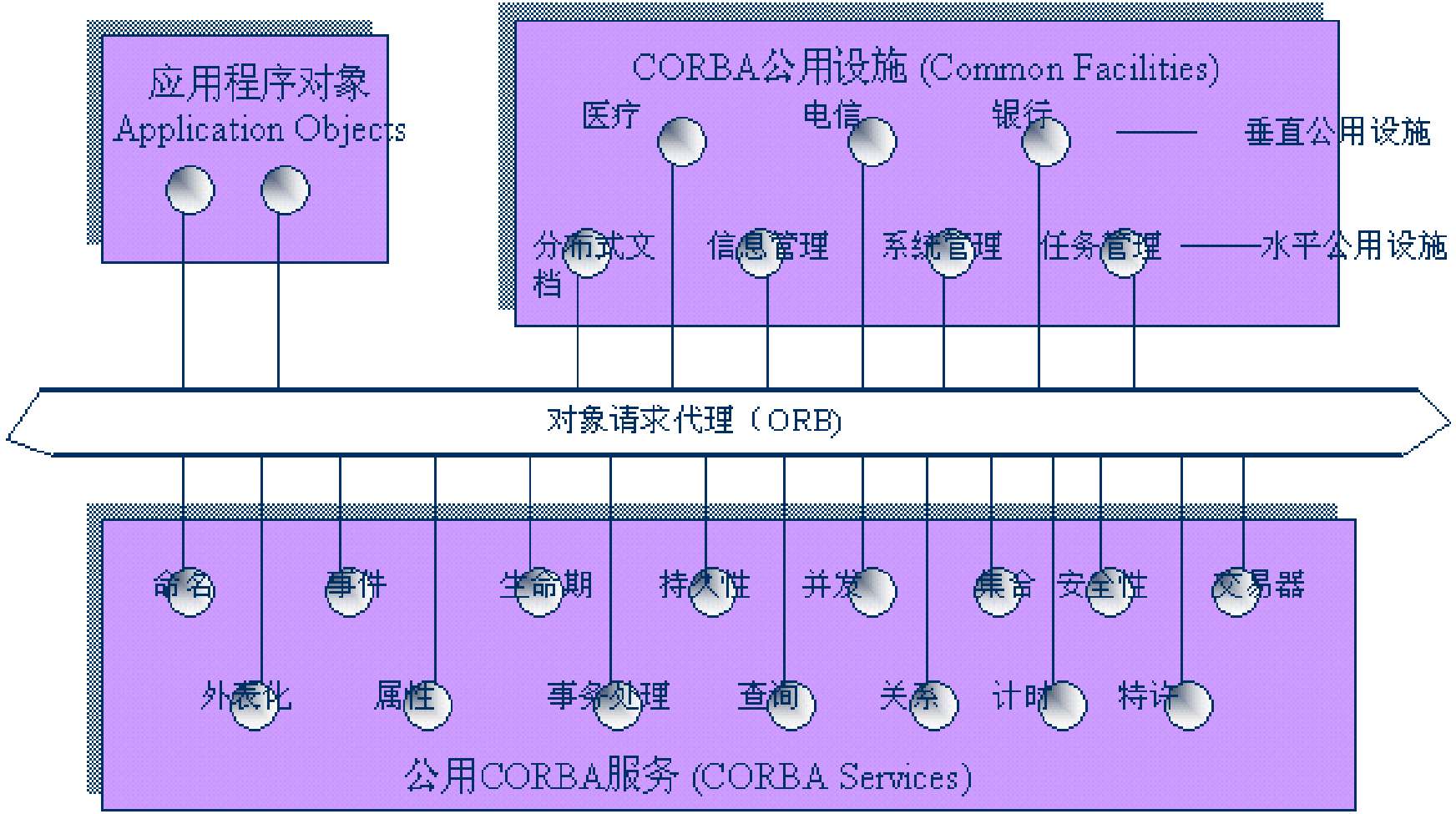
- 面向对象技术已成为软件系统构造的基本技术
- OO的精髓在于对象维护自身的状态并通过消息进行通信，这对于分布式计算环境是非常理想的
- 分布对象技术

  核心内容在于分布对象之间的互操作，尤其是异构环境中的互操作问题

典型的面向对象的分布计算环境

- CORBA：Common Object Request Broker Architecture
- DCOM：Distributed Component Object Model
- EJB：Enterprise Java Bean

CORBA


应用程序对象
Application Objects
CORBA公用设施 (Common Facilities)
医疗
电信
银行
垂直公用设施
分布式文档
信息管理
系统管理
任务管理
水平公用设施
对象请求代理（ORB)
命名
事件
生命期
持久性
并发
集合
安全性
交易器
外表化
属性
事务处理
查询
关系
计时
特许
公用CORBA服务 (CORBA Services)

分布计算环境的发展历程

- 经典的C/S技术
- 远程过程调用RPC技术
- 面向对象的分布计算环境
- 面向构件的分布计算环境
- 面向Web的分布计算环境
- 面向资源整合的分布计算环境

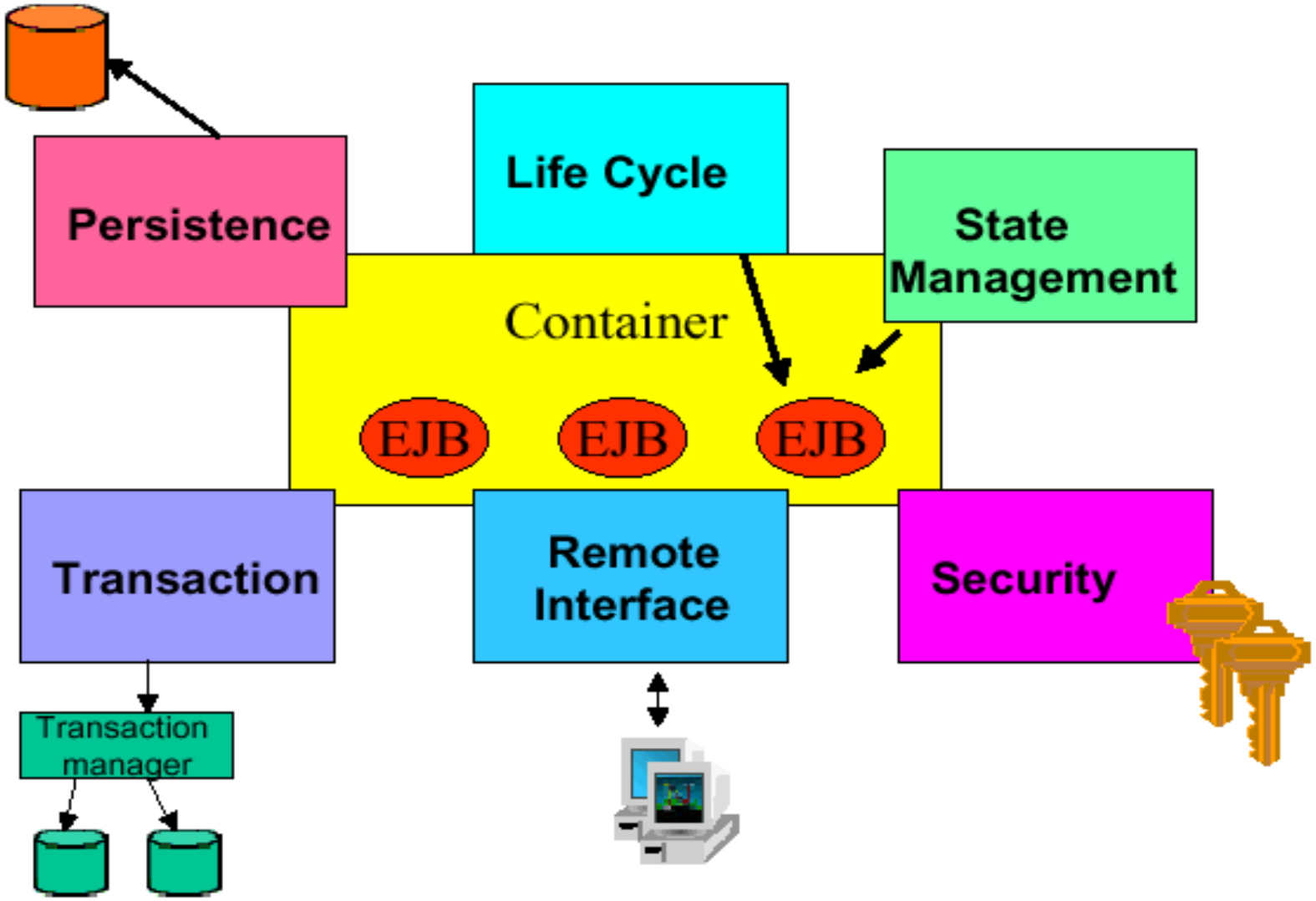
面向构件（组件）的分布计算技术

- 基于对象的应用程序--通过重用类库中的对象实现代码重用
  - 真正实现了信息隐藏和数据抽象的承诺
  - 各种应用程序分享细颗粒的对象（类库）
  - 对象仍然是被动的，仍然需要一种结构（控制流）将它们连接到一起
  - 不能把握将对象拼装在一起所需的逻辑联系

    ==> 不能提供高度的代码重用机制
- 大量的应用程序，特别是同一领域中的应用程序，分享相似的结构。这些结构并没有经过通常的面向对象技术而得到重用。
  - 解决方法：基于构件的软件体系结构

    ➔框架（+对象总线）、构件

例：EJB 服务器


Persistence
Life Cycle
State Management
Container
EJB
EJB
EJB
Transaction
Remote Interface
Security
Transaction manager

分布计算环境的发展历程

- 经典的C/S技术
- 远程过程调用技术
- 面向对象的分布计算环境
- 面向构件的分布计算环境
- 面向Web的分布计算环境
- 面向资源整合的分布计算环境

面向Web的分布式计算

- **Internet和Web技术的飞速发展和普及，对信息共享和应用协同提出了更高的要求**
  - **信息共享**
    - **Web 1.0**
    - **Web 2.0**
  - **广域网络环境下，异构应用程序的互操作**
    - **Web Service**

Web 1.0

- 以数据为核心，将以前没有放在网上的人类知识，通过商业的力量，放到网上去
- 通过WEB，互联网上的资源，可以在一个网页里比较直观的表示出来；而且资源之间，在网页上可以链来链去
  - 手工浏览互相链接的文档
  - 通过手工操作处理采购等商业事务
  - 下载文件
- 支持Web 1.0 的分布计算环境的主要组成及支撑技术
  - 浏览器、Web服务器
  - 搜索引擎、门户网站
  - HTTP、HTML
  - 静态网页技术、动态网页技术

Web 2.0

- 以人为出发点，让所有的人都忙起来，全民织网，然后用软件、机器的力量使这些信息更容易被需要的人找到和浏览。
  - 用户：贡献内容，传播内容，提供内容之间的链接关系和浏览路径
- 支持Web 2.0的主要组成及支撑技术
  - 支持 Wed 1.0的相关组成和技术
  - Blog： 用户织网，发表新知识，和其他用户内容链接，进而非常自然的组织这些内容
  - RSS： 用户产生内容自动分发，定阅
  - Podcasting： 个人视频/声频的发布/定阅
  - SNS： blog+人和人之间的链接
  - WIKI： 用户共同建设一个大百科全书
  - ……

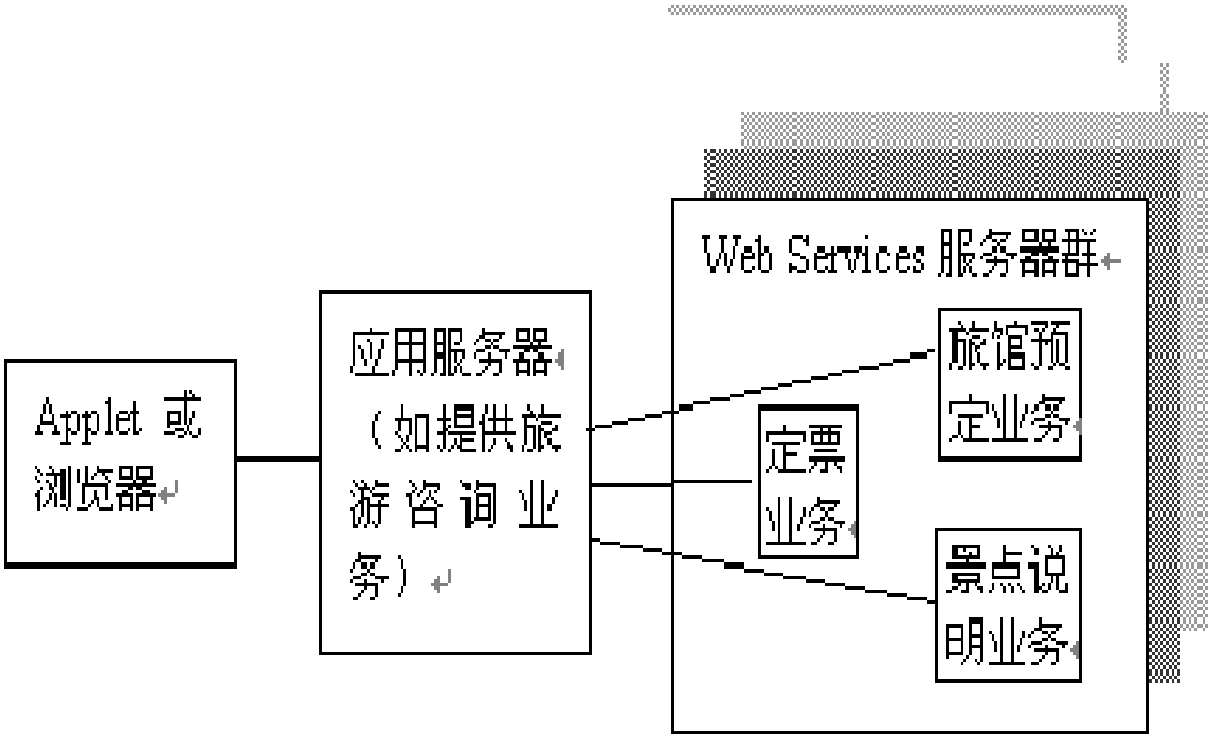
Web Service

- 传统的Web技术解决的问题是如何让人来使用Web应用所提供的服务，而Web Service则要解决如何让计算机系统来使用Web应用所提供的服务
- 目标：在现有的各种异构平台的基础上，构筑一个通用的，与应用无关、语言无关的技术层，各种不同平台之上的应用依靠这个技术层来实施彼此的连接和集成

基于XML以及
其他相关的Internet标准

什么是一个Web服务？

- 一个能够使用XML消息通过网络来访问的一个服务组件，它通过Interface描述了一组可访问的操作。（RPC Web Service）
- 通过对Web服务的集成，可以提供丰富的方便Web用户访问的应用
  - 面向服务的计算技术



分布计算环境的发展历程

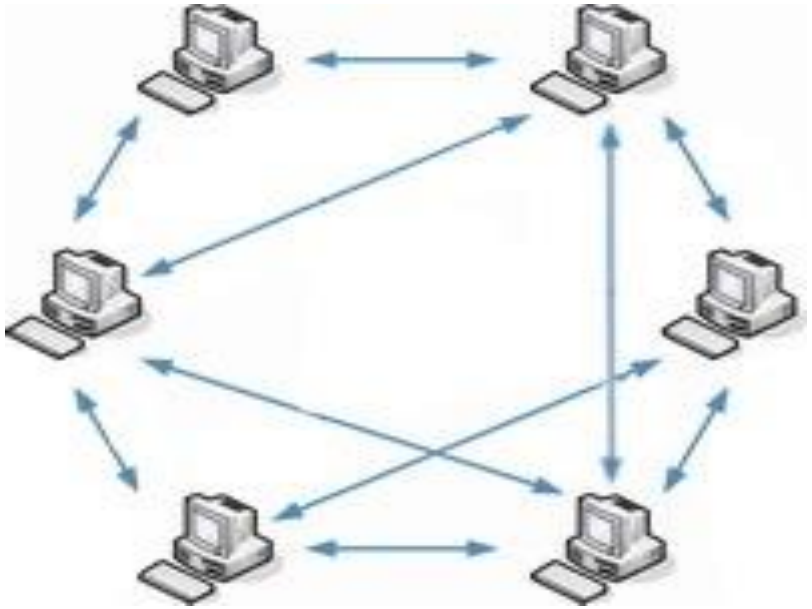
- 经典的C/S技术
- 远程过程调用技术
- 面向对象的分布计算环境
- 面向构件的分布计算环境
- 面向Web的分布计算环境
- 面向资源整合的分布计算环境

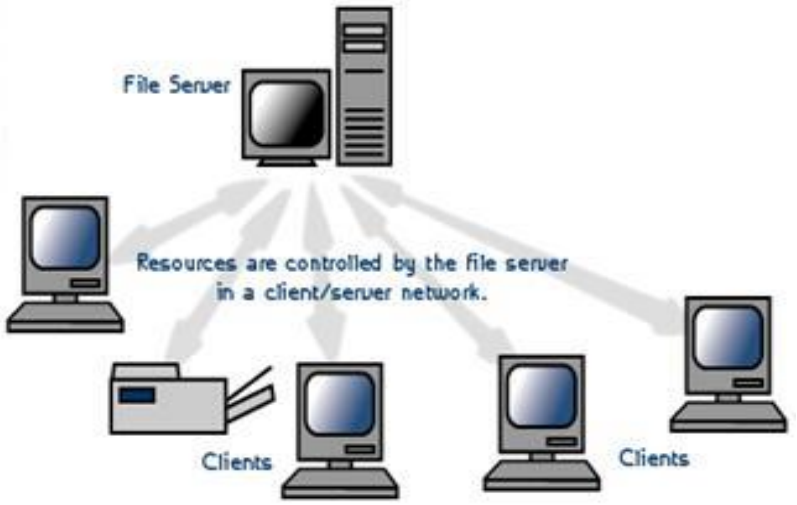
面向资源整合的分布式计算

- 随着Internet技术的飞速发展和普及，网络上分布着浩瀚的资源
  - 计算、存储、设备、服务、信息
- 另一方面，很多应用面临的问题越来越复杂，需要的资源越来越多
- 如何有效地整合资源
  - 资源共享、资源管理、协调工作
- 典型研究，大多以虚拟化技术作为基础，如
  - 网格计算
  - 云计算

P2P计算

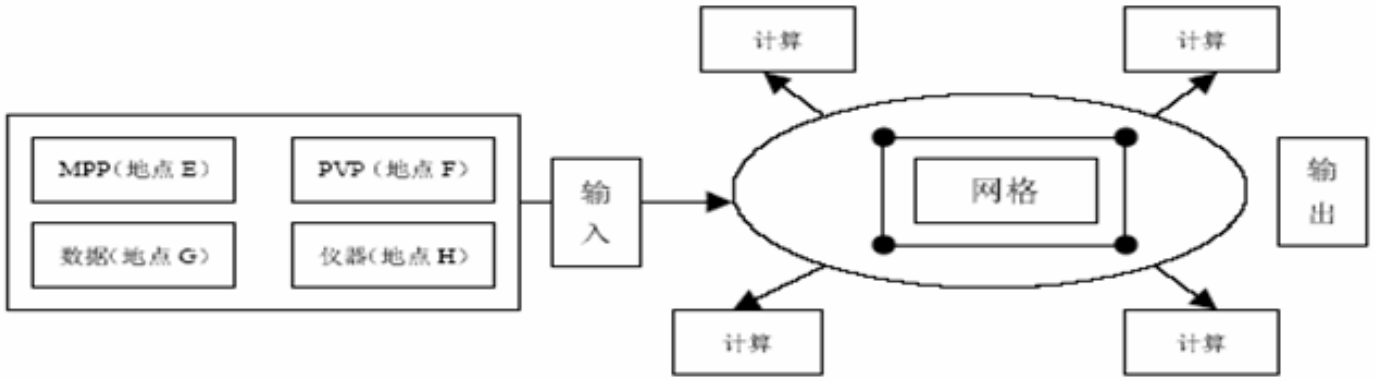
♦P2P 是一种分布式网络，网络的参与者共享他们所拥有的一部分硬件资源（处理能力、存储能力、网络连接能力等）和软件资源或者数据资源，这些共享资源能被其它对等节点（Peer）直接访问而无需经过中间实体。在此网络中的参与者既是资源提供者（Server），又是资源获取者（Client）





网格计算

- 把整个网络整合成一台巨大的超级计算机，实现计算资源、存储资源、数据资源、信息资源、知识资源等的全面共享
- 要解决的主要问题包括：
  - 如何共享各类资源，资源虚拟化技术
  - 然后在资源共享的基础上，通过各个资源的协作来完成各种任务
  - 提供透明的，方便的使用这些功能的接口

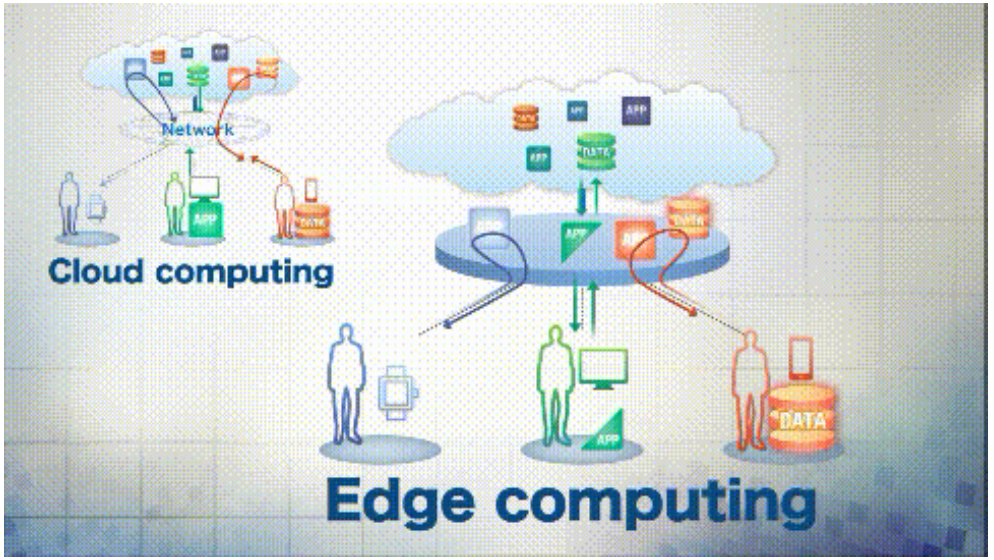


云计算

- 云计算是一种共享基础架构的方法，它可以将巨大的系统池连接在一起以提供各种IT服务（IBM）
- 使计算能力、存储能力、网络能力、各种软件服务能力等可以向电能一样提供给客户

边缘计算

- 边缘计算是指在靠近物或数据源头的一侧，采用网络、计算、存储、应用核心能力为一体的开放平台，就近提供最近端服务
  - 应用场景例：智能家居、智能交通、车联网、自动驾驶、CDN、……



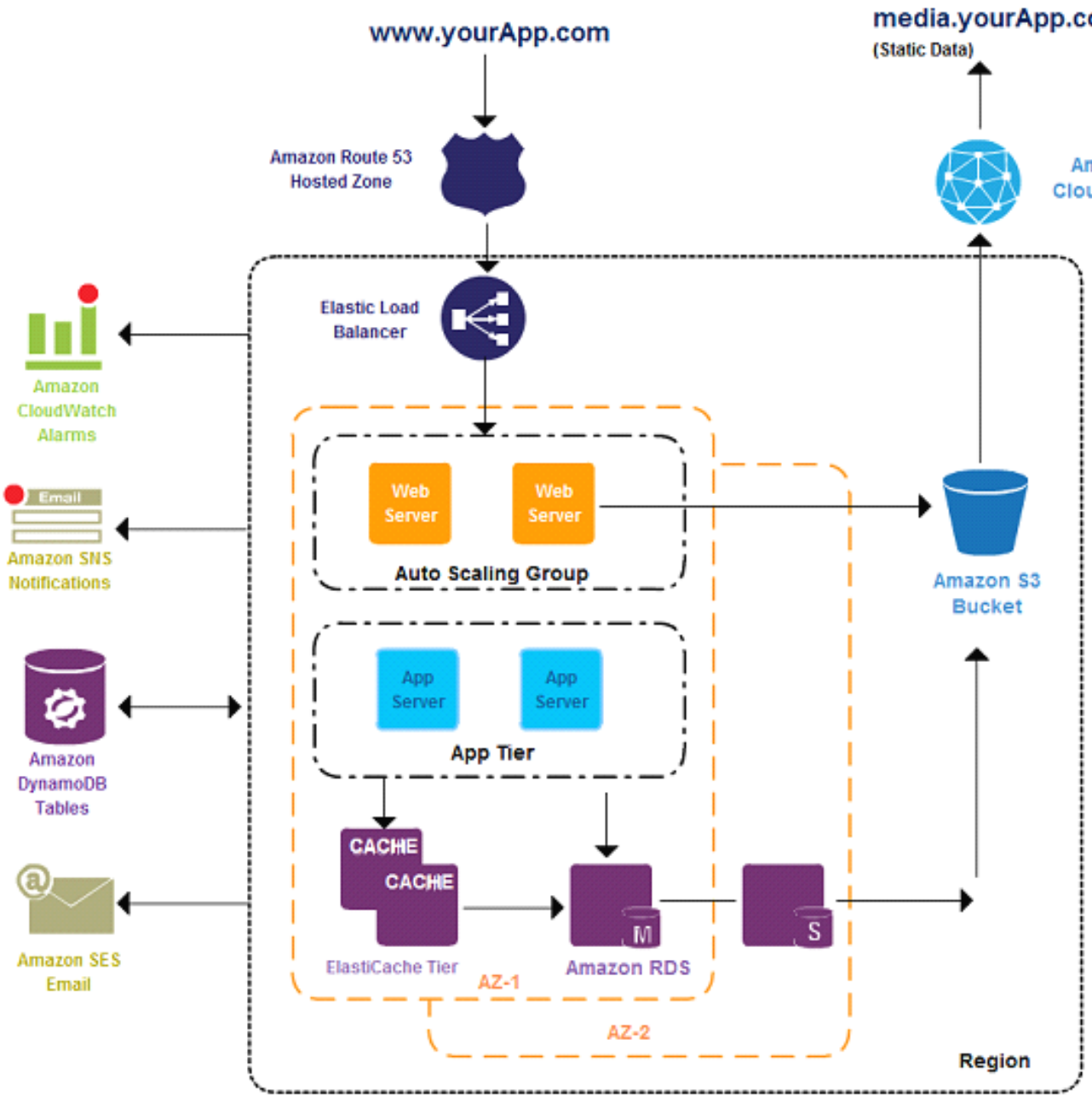
https://www.cnblogs.com/injet/p/9205386.html

新分布计算环境层出不穷

- 21世纪以来，随着众多互联网公司的崛起，用户量和数据量迅速增长，出现了很多新技术新环境，如：
  - Google的GFS，分布式文件系统
  - Google的Bigtable，NoSQL数据库系统
  - Google的Chubby,分布式协调系统
  - Google的MapReduce，并行数据处理框架
  - Apache的Spark，并行数据处理框架
  - Facebook的Thrift，分布式调用框架
  - 阿里巴巴的Dubbo，分布式调用框架
  - Apache的Kafka，分布式发布订阅消息系统
  - ……

各种分布计算环境之间有联系

- 在计算技术和网络技术发展的不同阶段，有相应的分布计算技术和对应的支撑环境，这是需求和技术双向驱动的
- 大多技术的发展阶段有交叠
- 大多技术之间有着千丝万缕的联系，如：
  - 有的技术基于其他一些技术发展
    - 如：CORBA的ORB，Java的RMI学习了DCE的RPC技术
    - 如：网格服务基于Web服务
  - 有的技术可出现在系统的不同层次中
    - 如：Web服务的内部实现可基于EJB、Spring
    - 如：云中部署着EJB、Spring服务器实例
    - 如：Kafka通过 Zookeeper 管理集群配置
- 由于遗留系统、市场占有率、面向的应用、新技术的成熟度等问题，新旧技术常常在很长一段时间内并存


3-Tier Auto-scalable Web Application Architecture
www.yourApp.com
media.yourApp.com
(Static Data)
Amazon Route 53
Hosted Zone
Amazon
CloudFront
Elastic Load
Balancer
Amazon
CloudWatch
Alarms
Email
Amazon SNS
Notifications
Web
Server
Web
Server
Auto Scaling Group
Amazon S3
Bucket
App
Server
App
Server
App Tier
Amazon
DynamoDB
Tables
CACHE
CACHE
ElastiCache Tier
M
Amazon RDS
S
Amazon SES
Email
AZ-1
AZ-2
Region

主要内容

- 分布式系统及其挑战
- 什么是分布式计算？
- 什么是分布计算环境
- 分布计算技术的发展历程
- 课程简介

主要内容

- 构造基于网络环境的大型软件系统时将遇到什么样的问题？现有的分布计算技术及支撑环境概述
  - 绪论
- 分布式系统的一些基本概念：进程、通信、命名、一致性与复制、容错、安全等
- 通过实例来讲述解决这些问题的原理和技术
  - CORBA
  - 基于构件的软件体系结构、JavaEE、EJB、轻量级框架
  - Web1.0、XML、Web 2.0、语义Web、Web Service
  - 网格、P2P、云计算、大数据、人工智能、边缘计算
  - GFS、Kafka、MapReduce、Spark……

参考资料

- 《分布计算环境》，王柏等， 北京邮电大学出版社
- 《分布式计算》，李文军等编著，机械工业出版社
- 《分布式软件体系结构》 （电子版），李文军等，中山大学计算机系
- 《分布式系统原理与范型》，辛春生等译，清华大学出版社
- 《分布式系统概念与设计》，金蓓弘、马应龙等译，机械工业出版社
- 《分布式系统设计实践》，李庆旭，人民邮电出版社
- 《分布式系统架构技术栈详解与快速进阶》，张程，机械工业出版社
- 参考国际国内相关领域的规范、著作、学术论文和其它网上资料等

考试方式

Assessment

平时：20％

Final Exam - 80% （开卷）

联系方式

- 邹华 ： zouhua@bupt.edu.cn

新科研楼519

61198133

说明

- **课程特点**
  - 内容比较分散，覆盖面广，概念多
  - 原理性和实践性都比较强
- **最好学过：**操作系统、计算机网络、某面向对象程序设计语言
- **几点建议**
  - 每次课后对课件进行复习，并与前面介绍的进行比较
  - 掌握各种分布计算环境的基本思想
    - ➔ 从技术出现的背景出发，多问自己：为什么这个技术出现了？它主要想解决什么问题？它为什么是这样的一种架构？主要特点是什么？
  - 带着比较的观点学习，分析各种技术的相同点和不同点
  - 选某个环境试用，亲身体验

多说几句

- **本课程的很多概念没有完全确定的定义**
  - 不同的定义在思想上大体相同，但在细节上或者在出发点上等，有着差异
    - ➔ 应该去比较，而不是试图找出一个完全正确的来
- **相关技术的分类方式也不统一**
  - 基于的分类标准不一样
- **要学会思考，形成自己的理解**
  - 此类专业课没有精准、完善的理论体系
    - ➔ 大多依赖讲授者自身的理解
  - 很多的东西，从不同的角度去看，得到的视图是不一样的，而且理解也可能是不一样的