



## 第4章 分布式软件体系结构风格

### 4.4 三层C/S体系结构 3-Tier C/S Architecture

---

刘其成

计算机与控制工程学院

ytliuqc@163.com

2018-09

## 两层C/S结构局限

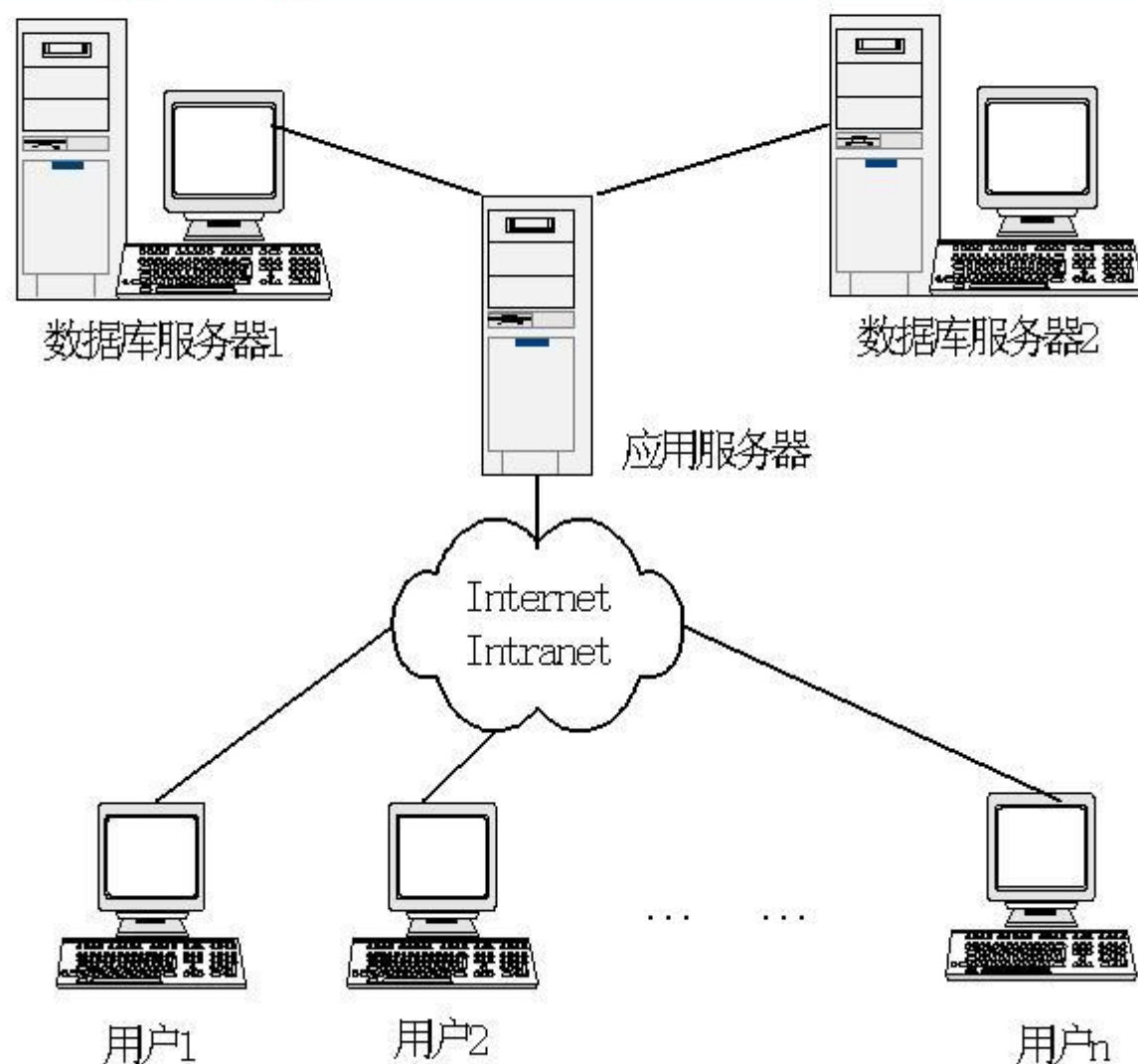
- 两层C/S体系结构具有强大的数据操作和事务处理能力，模型思想简单，易于人们理解和接受。
- 但随着企业规模的日益扩大，软件的复杂程度不断提高，两层C/S结构存在以下缺陷：
  - 两层C/S结构是单一服务器且以**局域网为中心的**，所以难以扩展至大型企业广域网或Internet。
  - 软、硬件的组合及集成**能力有限**。
  - **客户机的负荷太重**，难以管理大量的客户机，系统的性能容易变坏。
  - 数据安全性不好。因为**客户端程序可以访问数据库服务器**，那么，在客户端计算机上的其他程序也可想办法访问数据库服务器，从而使数据库的安全性受到威胁。
- 因为两层C/S体系结构存在缺点，三层C/S体系结构应运而生

## 三层C/S结构(3-Tier C/S Architecture)

- 三层C/S体系结构的出现克服了两层C/S的缺陷
- 在客户端与数据库服务器之间增加了一个中间层
  - 中间层可能为
    - 事务处理监控服务器、消息服务器、应用服务器等
  - 中间层负责
    - 消息排队、业务逻辑执行、数据传输等功能
- 通常，在三层C/S体系结构中，中间层增加的是一个应用服务器。



# 三层C/S结构(3-Tier C/S Architecture)



# 三层C/S结构 (3-Tier Client/Server)

- 基本组件:

- 数据库服务器

- 存放数据的数据库、负责数据处理的业务逻辑;

- 应用服务器

- 业务逻辑: 对数据进行处理;

- 客户机应用程序

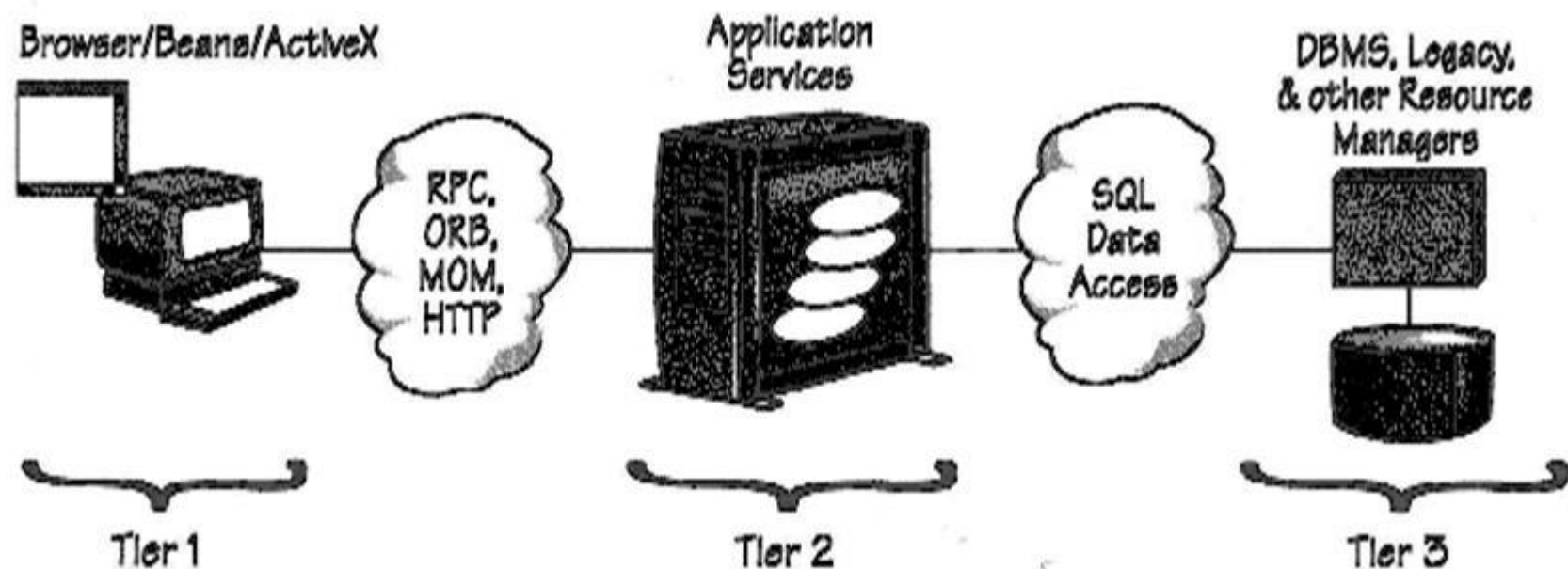
- **GUI:** 用户界面

## 三层C/S结构 (3-Tier Client/Server)

- 连接件：经由网络的调用-返回机制或隐式调用机制
  - 客户机 $\longleftrightarrow$ 应用服务器：客户机向应用服务器发送请求；应用服务器进行相关处理，将结果发给客户端；客户端接收返回结果。
  - 应用服务器 $\longleftrightarrow$ 数据服务器：应用服务器向数据服务器发送请求；数据服务器进行相关处理，将结果发给应用服务器；应用服务器接收返回结果。

# 三层C/S结构(3-Tier C/S Architecture)

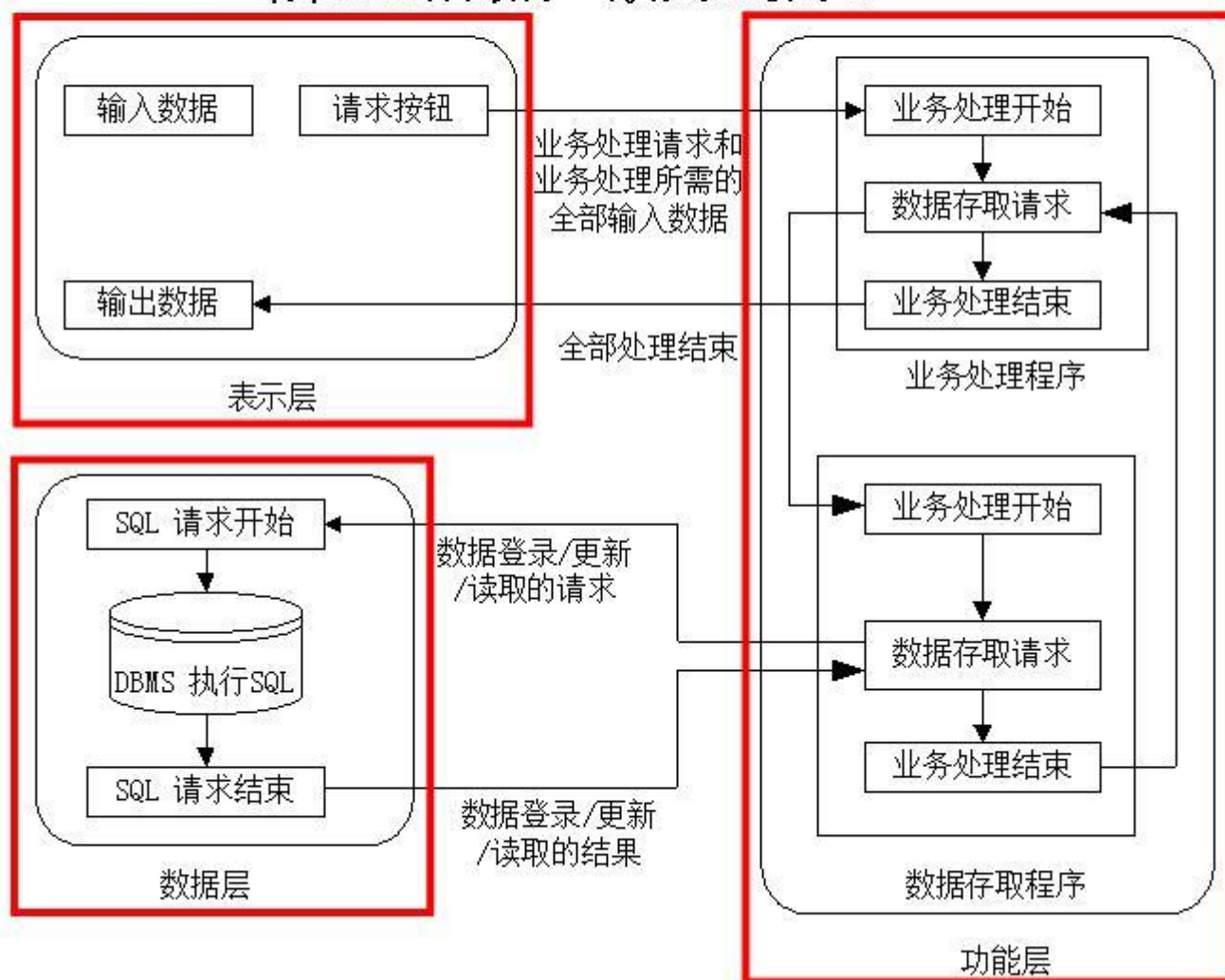
- 第1层：用户界面 **GUI**—表示层——客户机
- 第2层：业务逻辑—功能层——应用服务器
- 第3层：数据库—数据层——数据库服务器





# 三层C/S结构(3-Tier C/S Architecture)

## 三层C/S结构的一般处理流程





# 表示层

- 应用的**用户接口**部分，担负着用户与应用之间的对话功能；
- **检查**用户从键盘等输入的数据，显示应用输出的数据；检查的内容也只限于数据的**形式**和取值的**范围**，**不包括有关业务本身的处理逻辑**。
- 为使用户能直观地进行操作，通常使用**图形用户界面GUI**，操作简单、易学易用；
- 在变更时，只需要改写显示控制和数据检查程序，而**不影响其他层**；
- 不包含或包含一部分**业务逻辑**。

## 功能层(一)

- 应用系统的主体，包括**大部分业务处理逻辑** (通常以业务组件的形式存在，如**JavaBean/EJB/COM**等)；例如，在制作订购合同时计算合同金额，按照定好的格式配置数据、打印订购合同。
- 从表示层**获取**用户的输入**数据**并**加以处理**；
- 处理过程中需要从**数据层获取**数据或向数据层**更新**数据；
- 处理结果**返回**给表示层。

## 功能层(二)

- 用户检索数据时，要设法将有关检索要求的信息**一次性地传送**给功能层，而由功能层处理过的检索结果数据也**一次性地**传送给表示层。
- 通常，在功能层中包含有确认用户对应用和数据库存取**权限**的功能以及记录系统处理**日志**的功能。



# 数据层

- 数据库管理系统**DMBS**, 负责管理对**数据库数据的读写**
- 接受功能层的数据**查询请求**, **执行**请求, 并将查询结果**返回**给功能层;
- 从功能层接受数据**存取请求**, 并将数据**写入**数据库, 请求的执行结果也要**返回**给功能层。
- 数据库管理系统必须能迅速执行大量数据的更新和检索。现在的主流是**关系型数据库管理系统**, 因此, 一般从功能层传送到数据层的要求大都使用**SQL**语言。

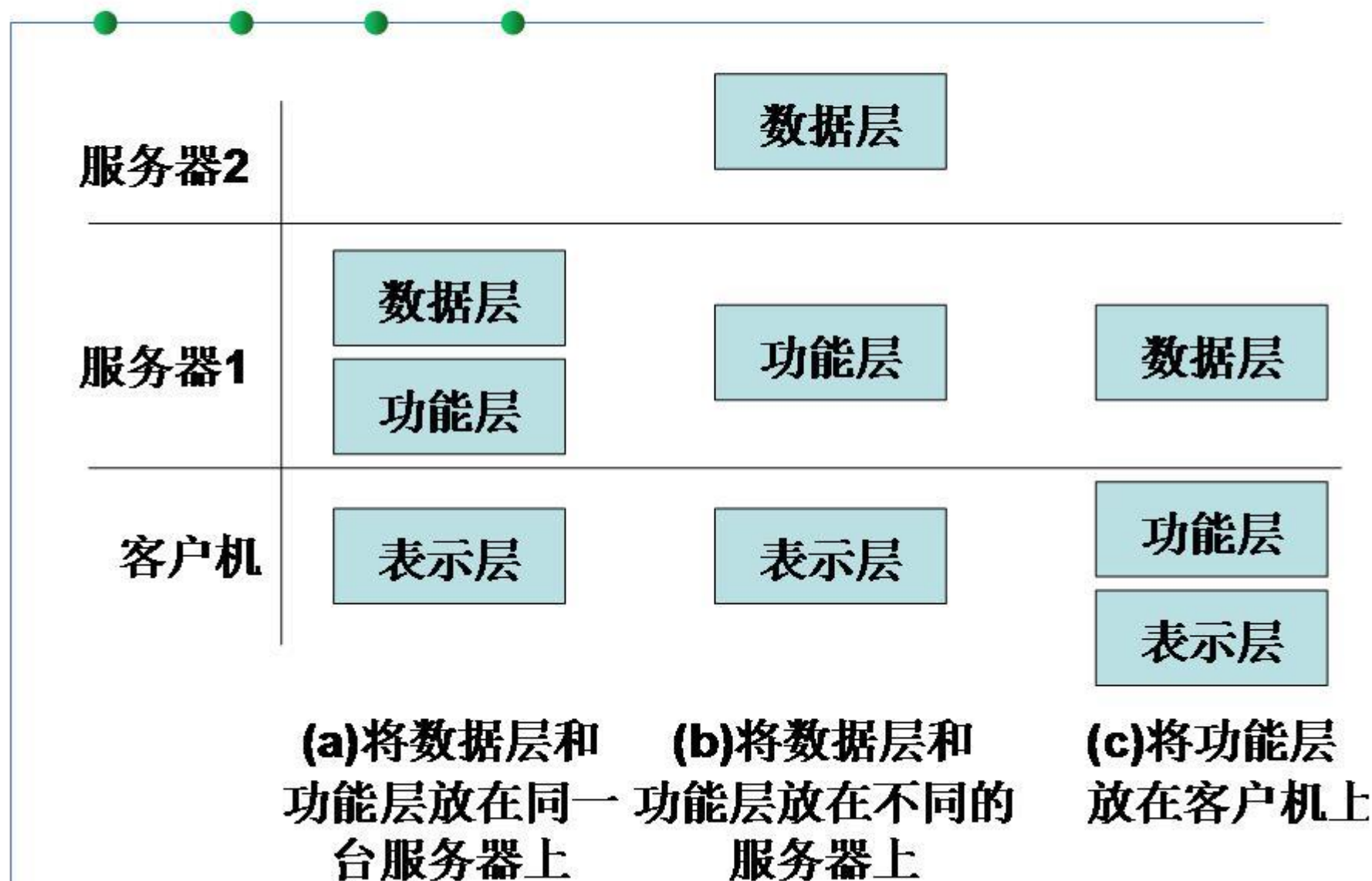
## 三层C/S结构的物理结构

- 三层C/S体系结构的解决方案
  - 对这三层进行明确分割，并在**逻辑上使其独立**
- 原来的数据层作为数据库管理系统已经独立出来(两层C/S已经将数据层分离出来)，三层C/S则要**将表示层与功能层分离开来**，形成独立的程序，并使二者之间的接口简洁明了。
- 问题：这三个层次在物理上是如何分布的？

- 一般情况是**将整个应用逻辑驻留在应用服务器上**，而只有**表示层存在于客户机上**。如图(a)或图(b)所示。这种结构被称为“瘦客户机”。
- 如果像图(c)所示的那样连功能层也放在客户机中，与二层**C/S**体系结构相比，其程序的可维护性要好得多，但是**其他问题并未得到解决**。客户机的负荷太重，其业务处理所需的数据要从服务器传给客户机，所以系统的性能容易降低。

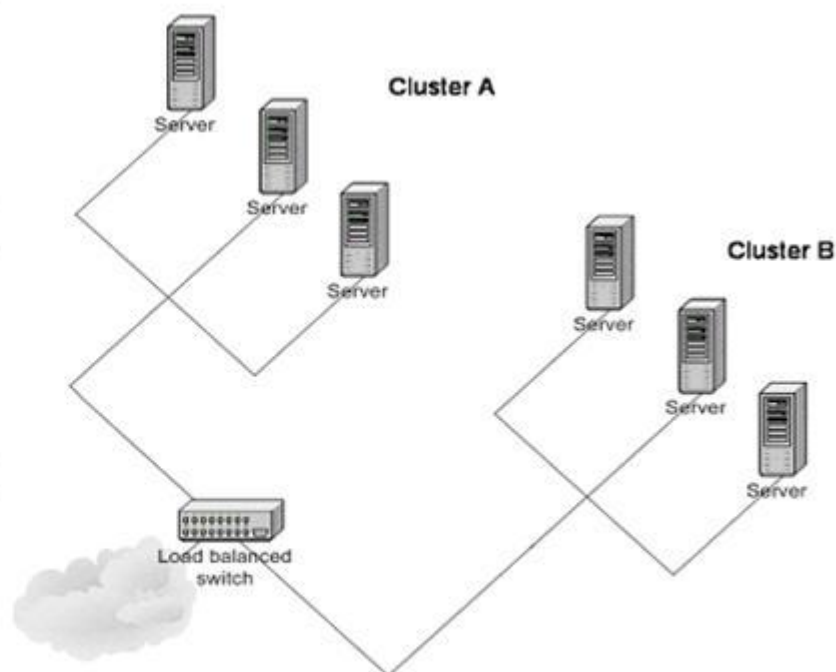
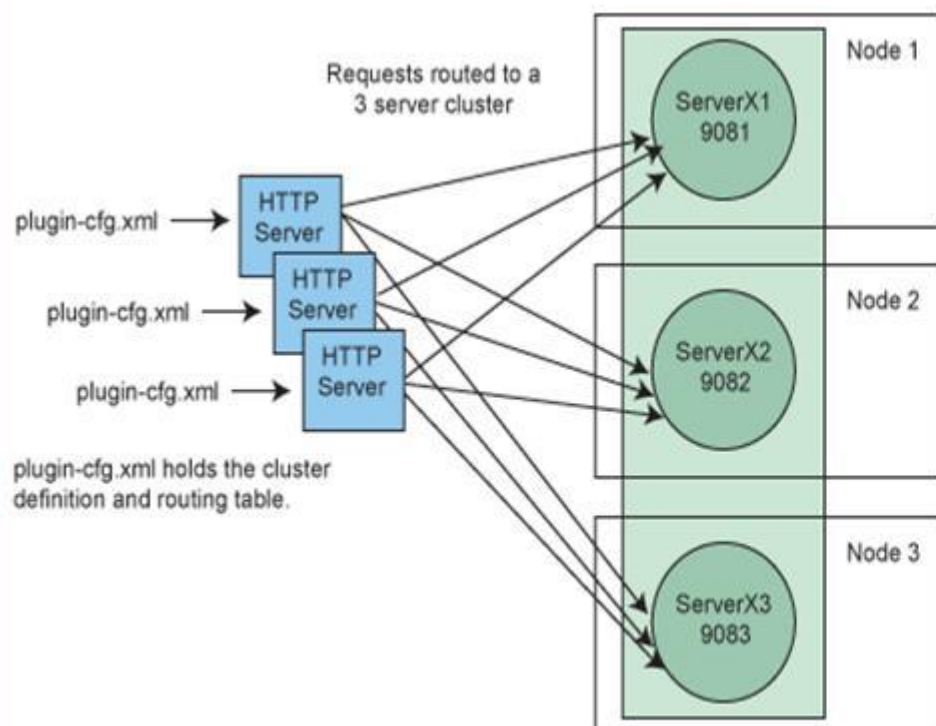


- 如果将功能层和数据层分别放在不同的服务器中，如图22(b)所示，则服务器和服务器之间也要进行数据传送。
- 但是，由于在这种形态中三层是分别放在各自不同的硬件系统上的，所以灵活性很高，能够适应客户机数目的增加和处理负荷的变动。
- 例如，在追加新业务处理时，可以相应增加装载功能层的服务器。因此，系统规模越大这种形态的优点就越显著。



# 基于集群(Cluster)的C/S物理分布

- 事实上，功能层通常不是只驻留在同一台服务器上，数据层也是如此；
- 如果功能层(或数据层)分布在多台服务器上，那么就形成了基于集群(Cluster)的C/S物理分布模式。





## 三层C/S结构的优点(一)

- 在用户数目较多的情况下，三层C/S结构将**极大改善性能与灵活性**（通常可支持**数百**并发用户，通过集群可达**数万**并发用户）；
- 允许合理地划分三层结构的功能，使之在逻辑上保持相对独立性，能**提高系统和软件的可维护性和可扩展性——UI、BL、DB可以分别加以复用**；
- 允许更灵活有效地选用相应的平台和硬件系统，使之在处理负荷能力上与处理特性上**分别适应于**结构清晰的**三层**；并且这些平台和各个组成部分可以具有**良好的可升级性和开放性**。

## 三层C/S结构的优点(二)

- 应用的各层可以并行开发，可以选择各自最适合的开发平台和开发语言。
- 利用功能层有效地隔离开表示层与数据层，未授权的用户难以绕过功能层而非法的访问数据层，为严格的安全管理奠定了坚实的基础。
- 将遗留系统（旧版本的系统）移植到三层C/S下将非常容易；

## 三层C/S结构的缺点

- 三层C/S结构各层间的通信效率若不高，即使分配给各层的硬件能力很强，其作为整体来说也达不到所要求的性能。
- 设计时必须慎重考虑三层间的通信方法、通信频度及数据量，这和提高各层的独立性一样是三层C/S结构的关键问题——分层风格的固有缺点



## 思考题

- 三层C/S体系结构：组件、连接件、工作机制、特点。



# 谢谢

---

**2018年10月29日**