深入理解 B/S 与 C/S 架构

一、什么是 C/S 架构

C/S 架构是第一种比较早的软件架构，主要用于局域网内。也叫 客户机/服务器模式。

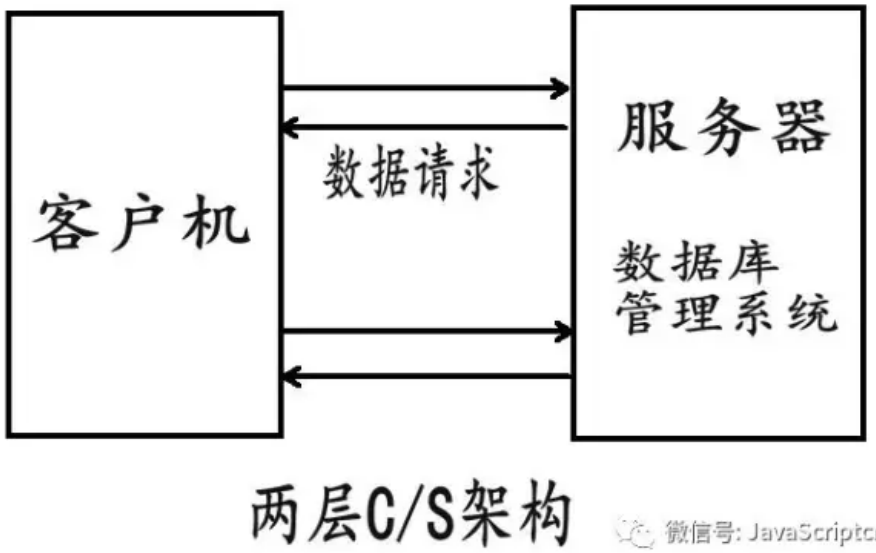
它可以分为客户机和服务器两层：

第一层：在客户机系统上结合了界面显示与业务逻辑；

第二层：通过网络结合了数据库服务器。

简单的说就是第一层是**用户表示层**，第二层是**数据库层**。

这里需要补充的是，客户端不仅仅是一些简单的操作，它也是会处理一些运算，业务逻辑的处理等。也就是说，客户端也做着一些本该由服务器来做的一些事情，如图所示：



两层C/S架构

C/S 架构软件有一个特点，就是如果用户要使用的话，需要下载一个客户端，安装后就可以使用。比如 QQ,OFFICE 软件等。

1、C/S 架构的优点：

1 C/S 架构的界面和操作可以很丰富。（客户端操作界面可以随意排列，满足客户的需要）

2 安全性能可以很容易保证。（因为只有两层的传输，而不是中间有很多层。

3 由于只有一层交互，因此响应速度较快。（直接相连，中间没有什么阻隔或岔路，比如 QQ，每天那么多人在线，也不觉得慢）

2、C/S 架构的缺点：

可以将 QQ 作为类比：

1 适用面窄，通常用于局域网中。

2 用户群固定。由于程序需要安装才可使用，因此不适合面向一些不可知的用户。

3 维护成本高，发生一次升级，则所有客户端的程序都需要改变。

二、什么是 B/S 架构

B/S 架构的全称为 Browser/Server，即浏览器/服务器结构。

Browser 指的是 Web 浏览器，极少数事务逻辑在前端实现，但主要事务逻辑在服务器端实现。

B/S 架构的系统无须特别安装，只有 Web 浏览器即可。

其实就是我们前端现在做的一些事情，大部分的逻辑交给后台来实现，我们前端大部分是做一些数据渲染，请求等比较少的逻辑。

B/S 架构的分层：

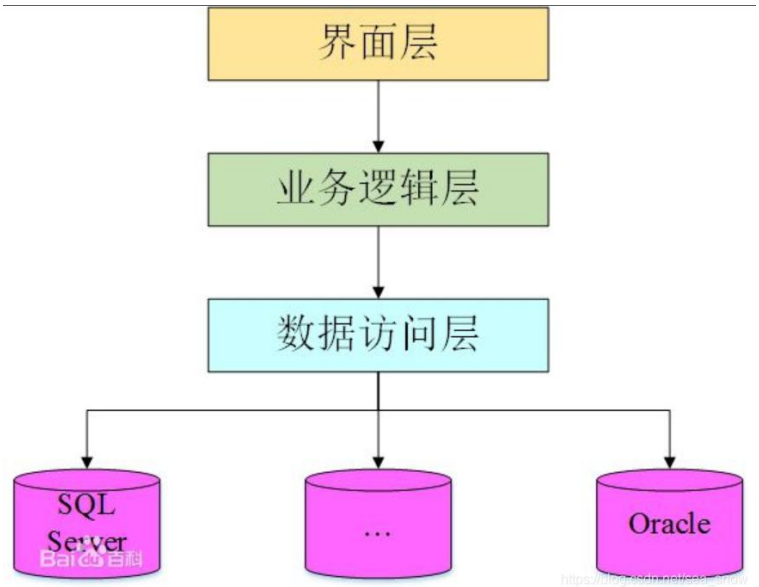
与 C/S 架构只有两层不同的是，**B/S 架构有三层**，分别为：

第一层表现层：主要完成用户和后台的交互及最终查询结果的输出功能。

第二层逻辑层：主要是利用服务器完成客户端的应用逻辑功能。

第三层数据层：主要是接受客户端请求后独立进行各种运算。

如图所示：



B/S 架构的优点：

1、客户端无需安装，有 Web 浏览器即可。

2、BS 架构可以直接放在广域网上，通过一定的权限控制实现多客户访问的目的，交互性较强。

3、BS 架构无需升级多个客户端，升级服务器即可。可以随时更新版本，而无需用户重新下载啊什么的。

B/S 架构的缺点：

1、在跨浏览器上，BS 架构不尽如人意。

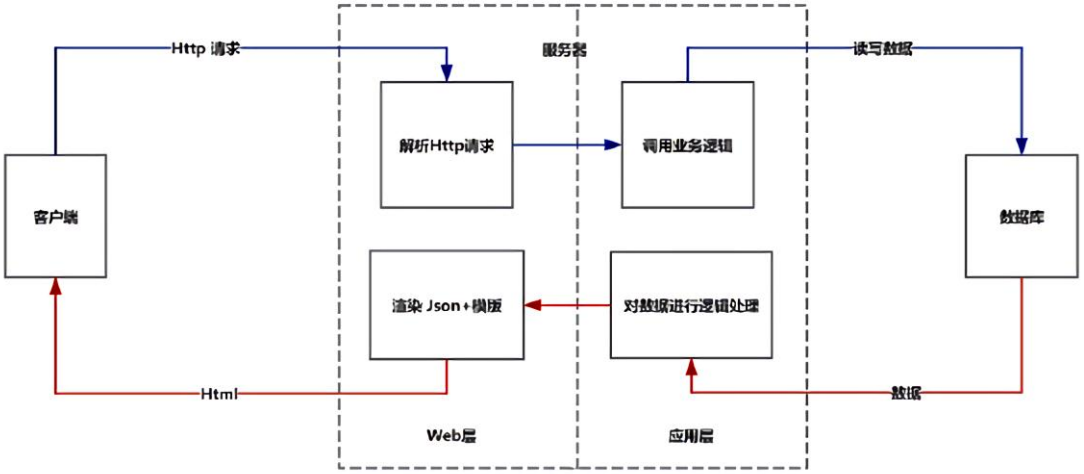
2、表现要达到 CS 程序的程度需要花费不少精力。

3、在速度和安全性上需要花费巨大的设计成本，这是 BS 架构的最大问题。

4、客户端服务器端的交互是请求-响应模式，通常需要刷新页面，这并不是客户乐意看到的。（在 Ajax 风行后此问题得到了一定程度的缓解）

三、B/S 架构的几种形式

第一种：客户端-服务器-数据库



这个应该是我们平时比较常用的一种模式：

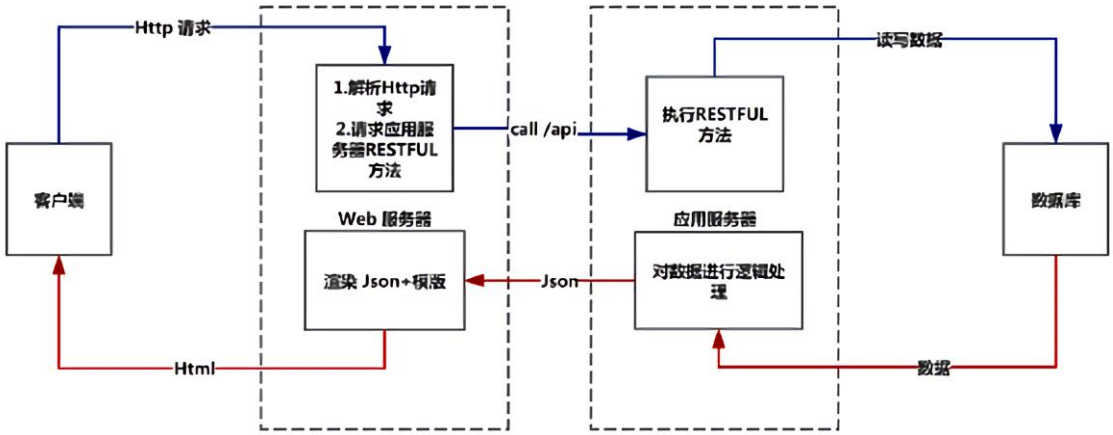
1、客户端向服务器发起 Http 请求

2、服务器中的 web 服务层能够处理 Http 请求

3、服务器中的应用层部分调用业务逻辑，调用业务逻辑上的方法

4、如果有必要，服务器会和数据库进行数据交换．然后将模版＋数据渲染成最终的 Html，返送给客户端

第二种：客户端－web 服务器－应用服务器－数据库



类似于第一种方法，只是将 web 服务和应用服务解耦

1 客户端向 web 服务器发起 Http 请求

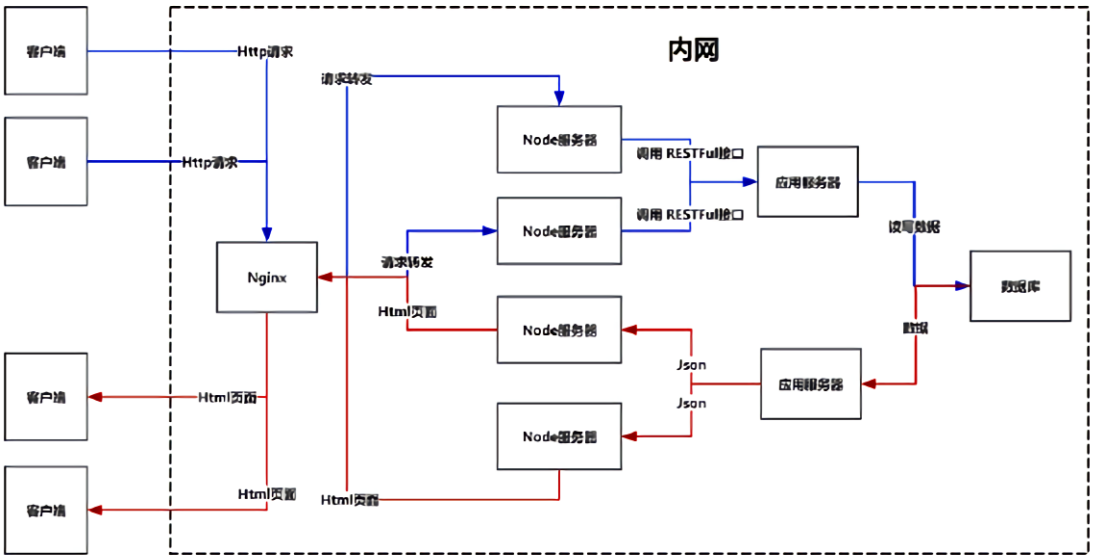
2 web 服务能够处理 Http 请求，并且调用应用服务器暴露在外的 RESTFUL 接口

3 应用服务器的 RESTFUL 接口被调用，会执行对应的暴露方法.如果有必要和数据库进行数据交互，应用服务器会和数据库进行交互后，将 json 数据返回给 web 服务器

4 web 服务器将模版＋数据组合渲染成 html 返回给客户端

第三种方法：客户端－负载均衡器(Nginx)－中间服务器(Node)－应用服务器－数据库

这种模式一般用在有大量的用户，高并发的应用中。



1、整正暴露在外的不是真正 web 服务器的地址，而是负载均衡器器的地址

2、客户向负载均衡器发起 Http 请求

3、负载均衡器能够将客户端的 Http 请求均匀的转发给 Node 服务器集群

4、Node 服务器接收到 Http 请求之后，能够对其进行解析，并且能够调用应用服务器暴露在外的 RESTFUL 接口

5、应用服务器的 RESTFUL 接口被调用，会执行对应的暴露方法.如果有必要和数据库进行数据交互，应用服务器会和数据库进行交互后，将 json 数据返回给 Node

6、Node 层将模版＋数据组合渲染成 html 返回反向代理服务器

7、反向代理服务器将对应 html 返回给客户端

Nginx 的优点有:

1、它能够承受、高并发的大量的请求，然后将这些请求均匀的转发给内部的服务器，分摊压力.

2、反向代理能够解决跨域引起的问题，因为 Nginx，Node,应用服务器，数据库都处于内网段中。

3、Nginx 非常擅长处理静态资源(img,css,js,video)，所以也经常作为静态资源服务器，也就是我们平时所说的 CDN

比如：前一个用户访问 index.html, 经过 Nginx－Node－应用服务器－数据库链路之后，Nginx 会把 index.html 返回给用户，并且会把 index.html 缓存在 Nginx 上，

下一个用户再想请求 index.html 的时候，请求 Nginx 服务器，Nginx 发现有 index.html 的缓存，于是就不用去请求 Node 层了，会直接将缓存的页面(如果没过期的话)返回给用户。

四、发展前景

1、 C/S 和 B/S 各有优势，C/S 在图形的表现能力上以及运行的速度上肯定是强于 B/S 模式的，不过缺点就是他需要运行专门的客户端，而且更重要的是它不能跨平台，用 c++在 windows 下写的程序肯定是不能在 linux 下跑的。

2、B/S 模式，它不需要专门的客户端，只要浏览器，而浏览器是随操作系统就有的，方便就是他的优势了。

而且，B/S 是基于网页语言的、与操作系统无关，所以跨平台也是它的优势，而且以后随着网页语言以及浏览器的进步，

B/S 在表现能力上的处理以及运行的速度上会越来越快，它的缺点将会越来越少。尤其是 HTML5 的普及，在图形的渲染方面以及音频、文件的处理上已经非常强大了。

不过，C/S 架构也有着不可替代的作用。