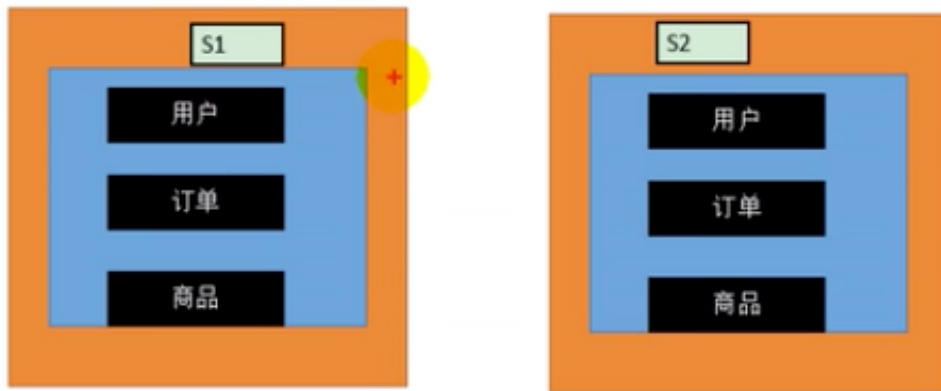


官网文档

1. 单一应用架构

即所有的功能都在一个项目中，新加入的功能也是在一个项目中，即使增加服务器也是在增加的服务器中加入整个项目

单一应用架构



- (1) 每次新功能都要发布打包，影响无关的功能
- (2) 不利于协同开发，有可能改用户和订单模块，但是改的是同一个项目
- (3) 单台服务器运行项目很大时，增加服务器是没有用的，对于每个服务器来说都是跑很大的应用。

2. 垂直应用架构

即将不同功能模块拆分开来，每个模块都是部署在不同的服务器上，每个模块是一个独立的从页面到数据库，如果单独一个模块压力比较大，那么为这个模块进行加服务器即可。

垂直应用架构



优点：

- (1) 利于协同开发，只开发对应的模块即可。
- (2) 利于部署，为单独模块的压力部署对应的服务器

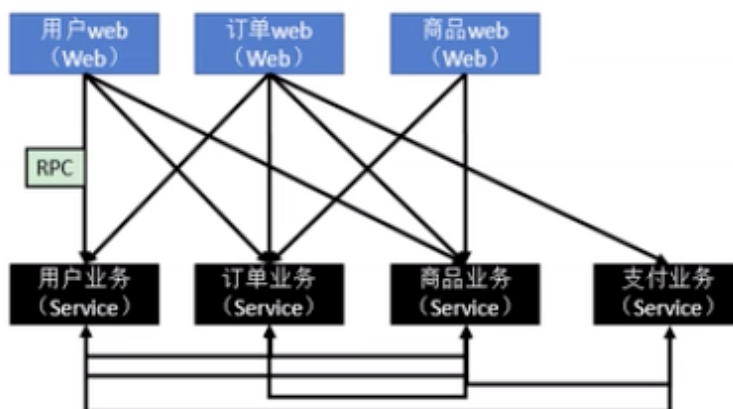
缺点：

- (1) 因为每个模块都是单独的页面，即页面和业务逻辑并没有分离，页面的改动是十分频繁的，所以每次页面改动就要造成发布
- (2) 不同模块之间也存在相互依赖，互相之间的调用非常麻烦。

3. 分布式服务架构

分布式服务架构

RPC：远程过程调用



将web和逻辑层分离，web和service部署在不同的服务器上，而之前因为在同一个服务器内，所以是走的进程内调用，但是现在在不同的服务器上，就引申出

rpc的概念。

而rpc的调用是分布式系统的关键。

4. 流动计算框架

分布式系统架构确实解决了调用之间的问题和相互之间的依赖问题。

但是会有有一个资源浪费的问题，比如用户业务没有压力那么大，但是部署了100台服务器，而订单业务压力很大，却部署了十台服务器。

所以需要有一个调度中心和管理中心去动态的扩容服务器，提高集群利用率也引入了soa的概念。

流动计算架构

