**作为一名架构师，我们要专业，要能看懂代码，即使光着臂膀去机房，也能独挡一面！即使同事搞不定问题，**

**或者撂挑子，你也能给老大一个坚定的眼神：不怕，有我在！还能在会议室上滔滔不绝，如若无人，**

**让不懂技术的妹子看你时眼神迷离，就好想落霞与孤鹜齐飞！**

**高并发、海量数据**

**什么是分布式？**

1. **任务分解**
2. **节点通信**

**分布式和集群的关系？**

**电商平台： 用户、 商品、订单、 交易**

**分布式： 一个业务拆分成多个子系统，部署在不同的服务器上**

**集群： 同一个业务，部署在多个服务器上**

# 计算机的发展历史

1. **1946 情人节**
2. **1964 IBM SYSTEM/360**

**超强的计算能力、 高可靠性**

1. **X86 CPU**
2. **RISC CPU 小型机**

**软件架构往集中式发展， 成为当时软件架构的主流**

# 分布式架构的发展

1. **时机成熟了**

**PC机的性能不断提升**

1. **企业必须要做**

**阿里去IOE运动：**

**IBM小型机 、 Oracle Rac、 EMC存储设备**

**PC mysql mariadb**

**2013年5月17号，最后一台IMB小型机下线**

**单机计算机的架构->分布式计算机架构**

# 架构的发展演变过程

**lamp**

**BAT**

# 什么是大型网站

1. **访问量（tps、qps）**
2. **数据量（存储数据量）**

**tps(transactions per second) : 每秒传输事务处理的个数,消息的入和出加上数据库的访问**

**qps(query per second): 每秒查询率是对一个特定的服务器在规定时间内所处理流量多少的衡量标准,机器的性能常用qps来进行衡量**

**.**

# 电商平台的发展

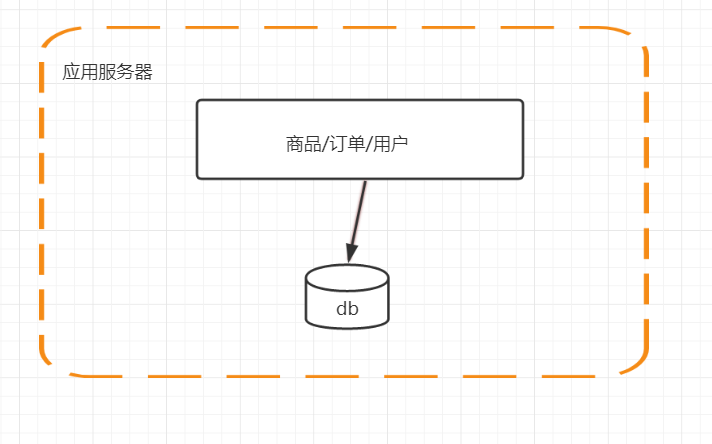
**用户、 商品、订单**

**容器： tomcat ；jsp/servlet**

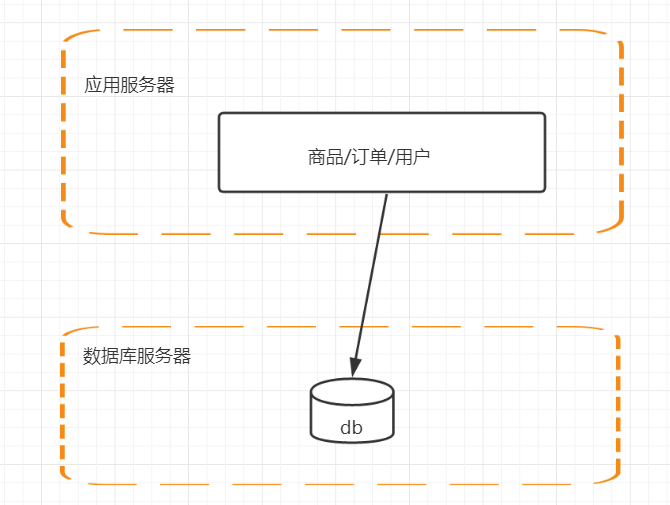
**数据库存储： mysql**

**网络层面的知识：tcp/udp**

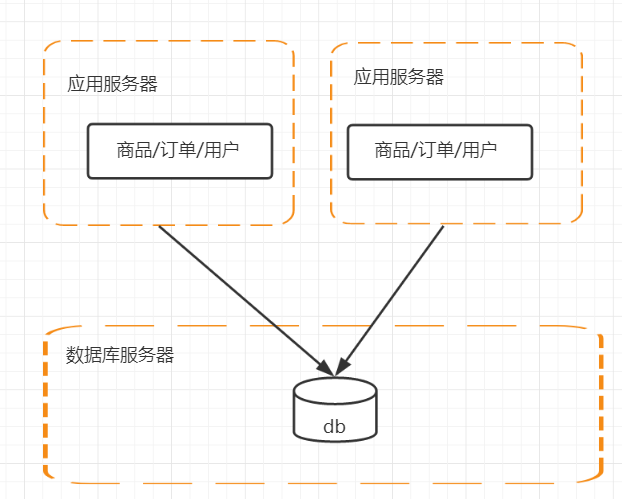
**第一版应用**



**第二版 单击负载越来越高，数据库服务器和应用服务器分离**



**第三版 应用服务器做集群**



**1 .session**

**cookie 存储的是JSESSIONID**

**ConcurrentMap key JSESSIONID values session**

**session跨域共享解决方案**

1. **session sticky**

**是指让同一客户端的请求，落在同一台服务器上，因为不会落在别的服务器上，所以自然就不会出现跨域问题。但是这个方案的缺点非常的明显，就是不管比采用什么算法，用户的请求落在哪一台服务器上都是由用户来决定的，可能会造成单点压力，并且如果一台服务器出问题，可能会造成一片区域的人无法访问。**

1. **session replication**

**是指服务器之间互相同步session信息，也就是说每台服务器上都保存着所有的session信息。这样做的缺点也是非常明显的。上文提到过，session是存在内存中的，会严重影响服务器性能，当然，你也可以把他存在数据库中，但是这会大大影响响应速度。还有一个缺点就是，当访问量过大时，由于互相同步的问题，会造成大量的网络开销**

1. **session 集中存储**

**是指把Session集中存储在一个第三方的服务器中，可以是Redis,可以是数据库或是其它什么东西。当需要访问的时候，都去这个服务器去查。这样做也有不小的缺点，首先是单点问题，如果这个服务器宕机，那么所有的服务都是不可用的，所以这里必须做集群，会浪费服务器资源。还有一点是，每次验证都需要来这个服务器来查，会凭白增加一次网络开销，降低访问速度**

**存储在db、 存储在缓存服务器 （redis）**

1. **cookie (主流) jwt(json web token)**

**access\_token(userid/token/timestamp)**

**soa架构和微服务架构**

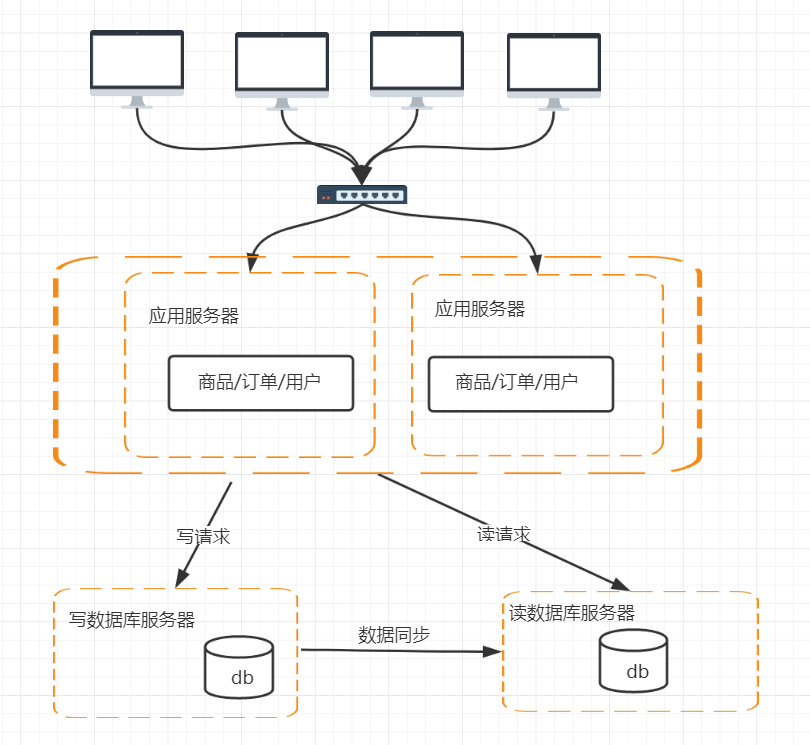
**状态信息不再保存在服务端，而是保存在客户端，客户端每次访问服务器的时候，把这个信息带给服务器。但是Cookie也有不少问题，最被人关注的就是安全问题，因为信息是保存在客户端的，就比较容易被人盗取、篡改，当然这些安全问题都是有解决方案的，这不是限制Cookie的主要原因，真正限制Cookie的原因是很多设备\*\*不支持\*\*Cookie**

1. **token**

**和Cookie类似，Token也是由客户端来维持状态的，信息存储在客户端内，具有平台无关性。Token实质上是服务端给客户端的一个字符串，上面包含着一些验证信息，相当于一个身份令牌，你拿着这个令牌就能得到他的服务。相比较于Cookie，Token更加的灵活，可以在任何地方生成，基于Token的权限系统是非常容易实现的**

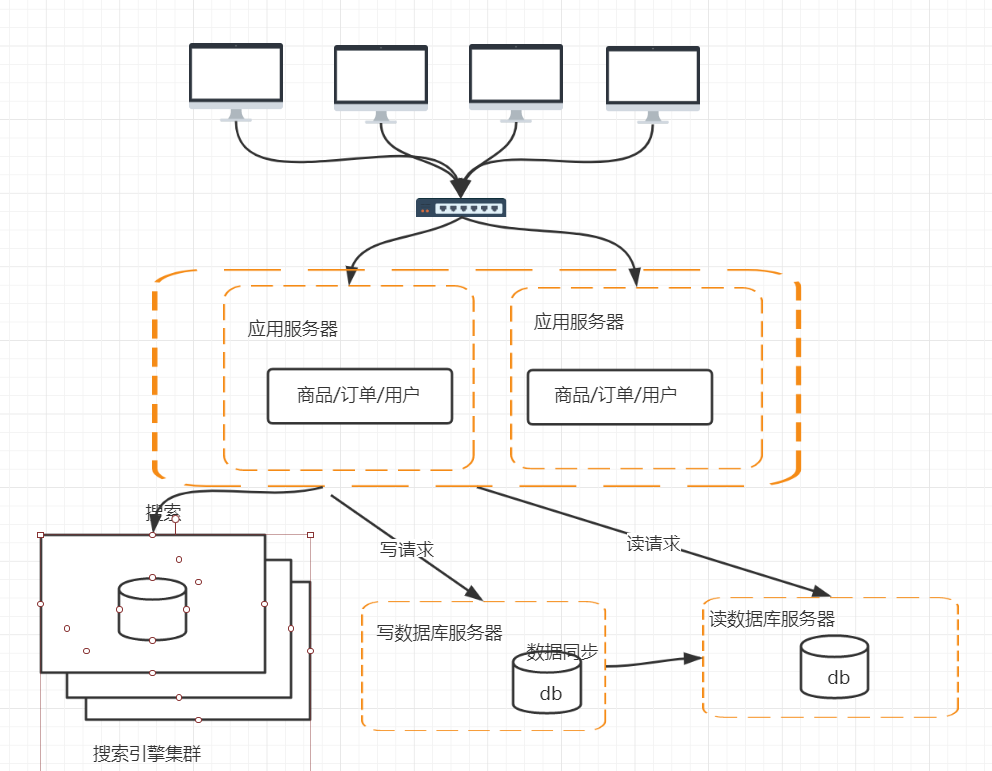
1. **如何做请求转发**

**第四版 数据库的高性能操作**



1. **数据库读写分离怎么操作**
2. **数据库的数据同步**
3. **数据库路由 mycat**

**电商平台最多的操作： 搜索商品**



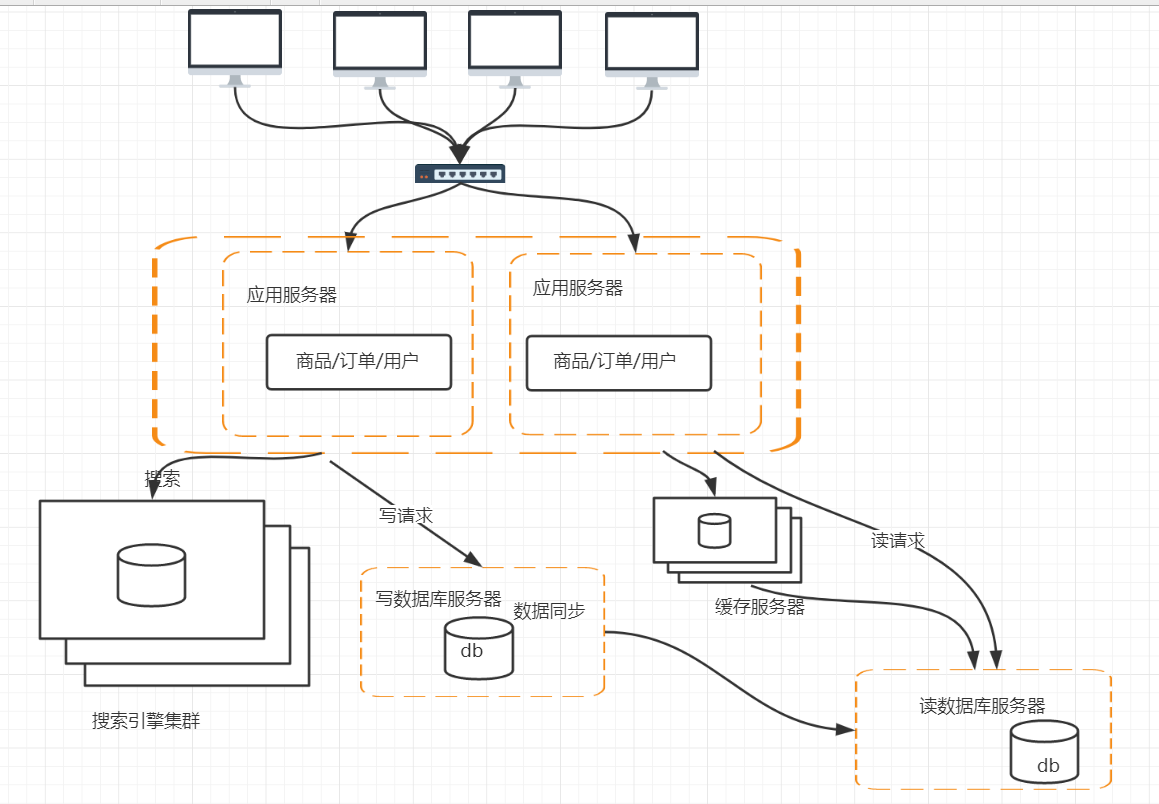
**问题：**

1. **搜索引擎的索引数据怎么去做同步，实时增量同步？ 还是定时全量同步？**

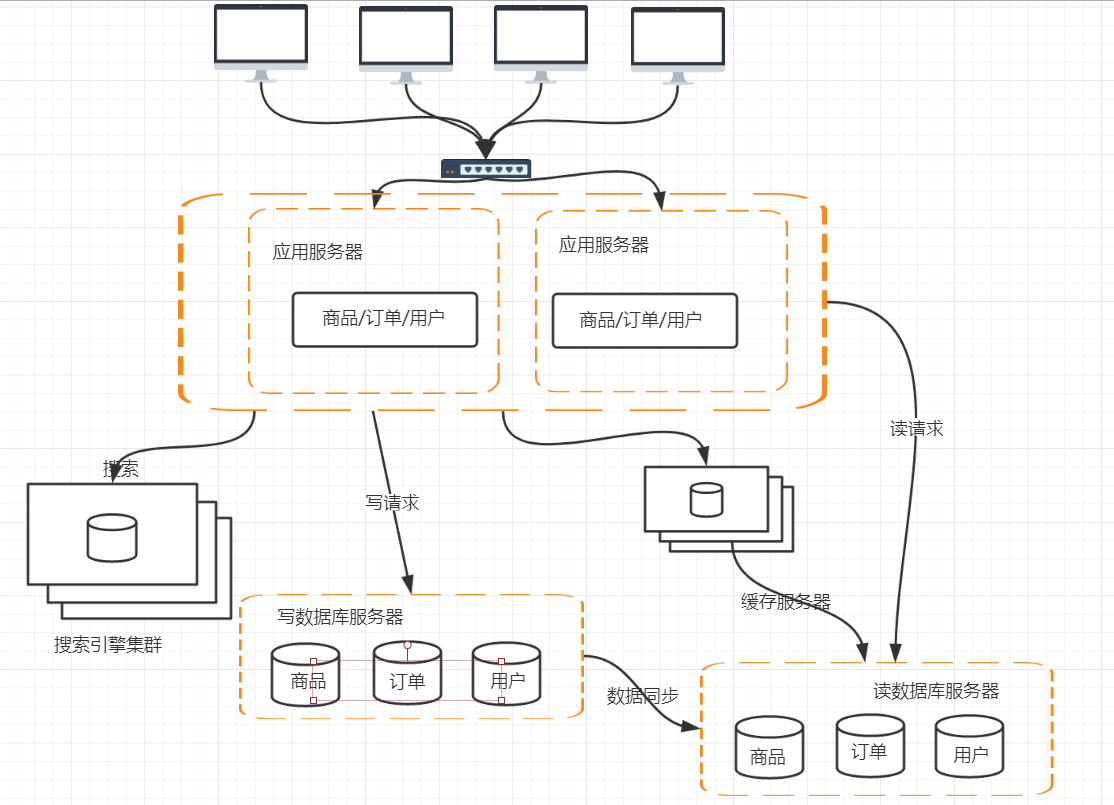
**第六版 解决访问量持续增高，引入缓存机制**

**用户量是没有上限的**

**缓存、 限流、 降级**



**第8️版本 数据库的水平/垂直拆分**



**第9版本**

