一、java语言的主流方向

java编程里，两个高峰：高并发/高可用

java语言 ==== java架构，

二、计算机与互联网的发展

apache 第web服务器：静态网页服务器----1995年。

lamp组合，开启互联网的大门。

三、互联网的行业变迁

web1.0：少数人员发布信息，大众接收信息。资讯类

web2.0：rss订阅。用户和服务器有互动。

客户端与服务器的交互模式：

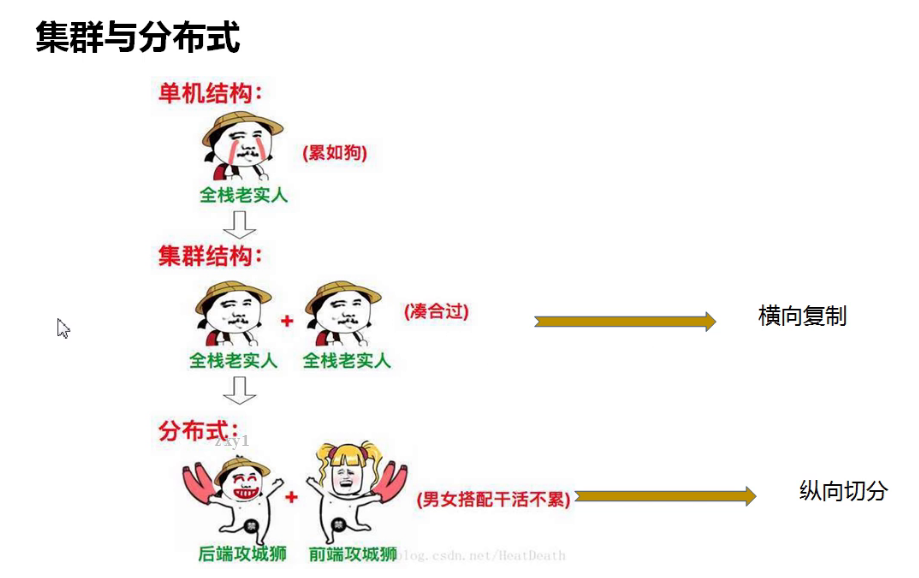
b/s结构，服务方式。

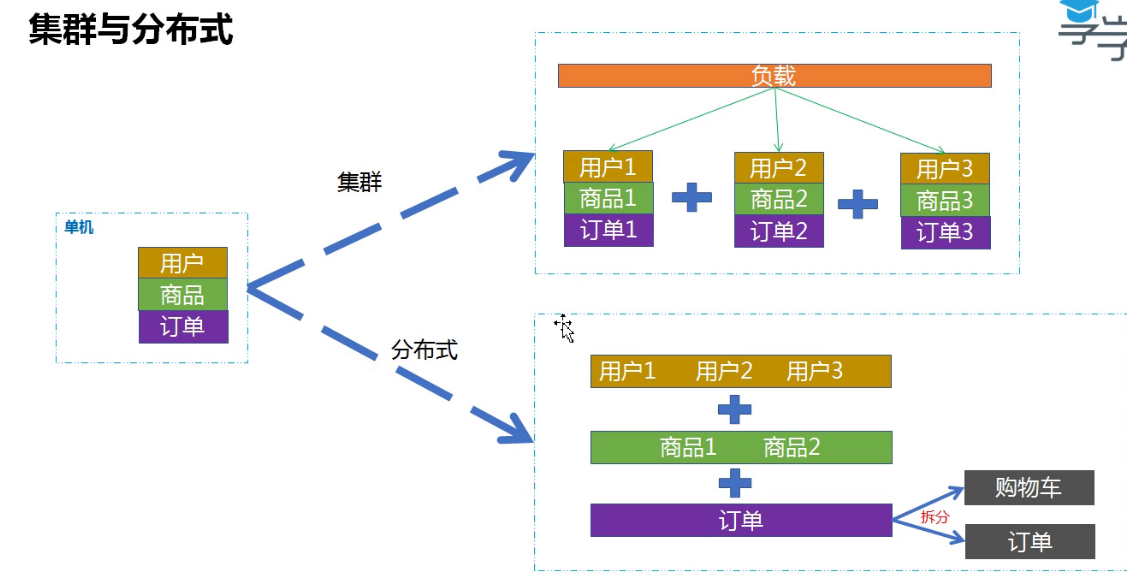
c/s方式：

四、集群与分布式的区别

集群：复制模式，每台机器做一样的事情。

分布式：两台机器分工合作，每台机器做的事情不一样。 业务流：A-->B--->C





五、浏览器请求服务器的过程

1、先请求DNS解析域名得到ip，再通过ip请求目标服务器。

例如：

更改本地host：-------》达到更改目标机器：欺骗浏览器，让以为我指定机器是DNS机器

自己做试验：

HOTS： 127.0.0.1 www.baidu.com

2、浏览器渲染出一个页面的过程：

浏览器----》解析本地html文件

解析服务器返回的网站内容。

对浏览器解析渲染页面来，上述两种方式得到的结果，是没有区别的。

六、网站开发的模式变迁

1、数据库开发网站：

数据库表user -----》 html模板里。

jsp开发：页面级/request/session/全局。 -------》 jsp页面 ${name}

2、数据库的操作，主要：增删改查。

主要压力来自查询：多个表联合，数据量膨胀。

如： select \* from A,B.

A表 B表

1 4

2 5

2 6

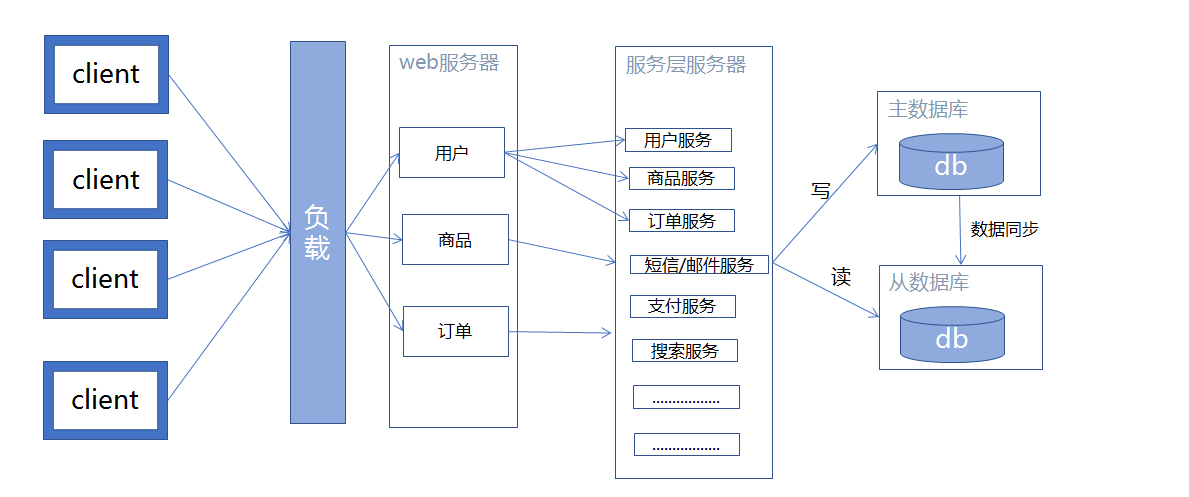
上述的一条sql，其结果是A/B两表记录的笛卡尔乘积。

此时，A表若新增一条数据，整体查询计算量是几何级上升的：查询量 = A 表量 \* B表量

3、 服务器的集群拓展

如果商品服务抗不住，----》集群解决：把整个服务复制一份。

七、java项目的分布式的拓展方案：

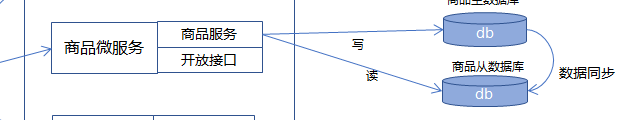


1、早期是一个war包，现在把service层压力大的service独立成单独出来，成为独立服务部署（jar包启动）

2、单独出来的服务，可以以集群模式复制拓展，这样的复制性价比较高。

3、大量的服务被独立出来后，这些独立的服务，就成为一个服务层网络。

八、微服务模式的拓展：



1、早期是一个war包，现把商品服务的control和service和dao全拆出，独立为一个服务。商品表也从数据库独立出来，成为一个独立商品数据库。

2、此时的商品服务，能够直接接收http请求，为前台服务。这样的一个完整服务，称之为微服务。

3、微服务易于上架下架，而不与其它模块逻辑迁连。

九、服务治理：

1、无论是分布式拆分，还是微服务拆分，最后形成的服务层应用，都需要相互之间调用。

2、这种服务间调用是跨系统的。跨系统调用rpc有很多协议： RMI，webservice，http请求，网络达到服务间访问。

dubbo：很多，dubbo协议

springcloud：使用http协议通信

3、服务集群，有负载路由。需要服务发现--zookeeper。

4、数据积累大：订单服务---积累。

-------分库/分表 -----需要修改业务sql语句。（有侵入性）-----------》mycat中间件

+ /分区。 -------》对sql语是没有侵入性。------------》对库没办法减压。

十、架构一个系统依据：

1、系统目标是并发数（tps）多少：tomcat的安全连接数在500tps上下

2、系统要承受的数据量级：mysql --- 单表到达 700W，性能会急剧下降。

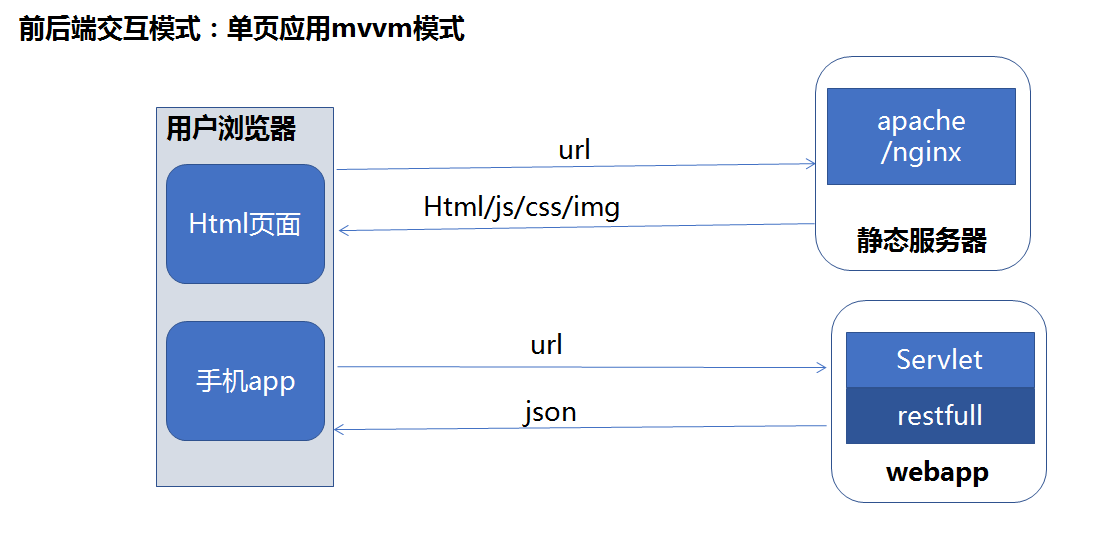
3、并发数（tps） transation per seconds：每秒钟的访问---每秒处理事务数

十一、前端发展：

1、一个url ---------------- 一个html页面 ---------》点击按钮跳转 --------》页面跳转。

2、页面，ajax方式刷新页面。

3、mvvm的开发流程



a、前端人员开发的程序包放在静态服务器里

b、浏览器访问静态服务器，得到前端html

c、html页面发起ajax请求到后台服务器，得到业务数据，渲染出页面。

十二、全网站搜索：

不指定业务表的搜索，任何一个中数据 like 匹配。

es/solr搜索中间件。

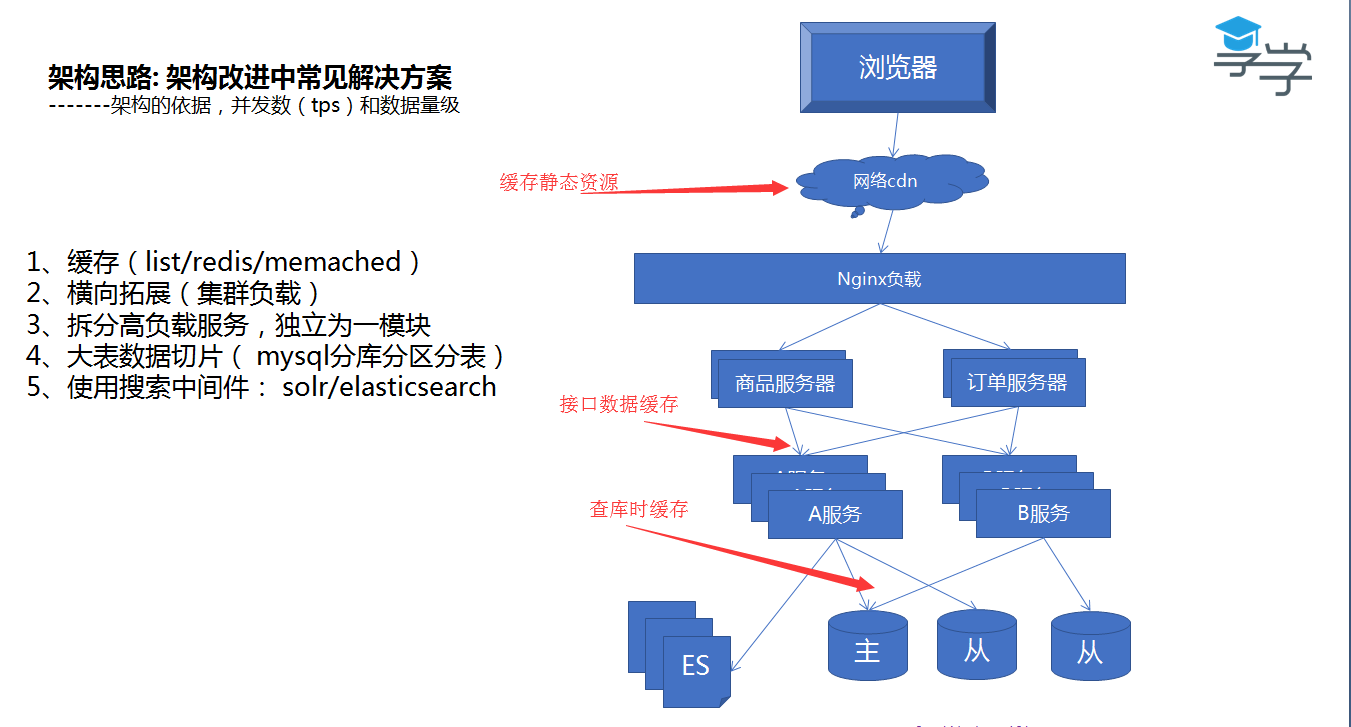
十三、架构方案：

1、首先考虑缓存方案解决性能问题

2、集群模式------比较重的模式：浪费成本

3、用分布式

十四、缓存解决方案



a、缓存服务器：redis、memcache

十四、集群引起的负载问题方案，如何session共享：

1、tomcat自己插件，能够session共享

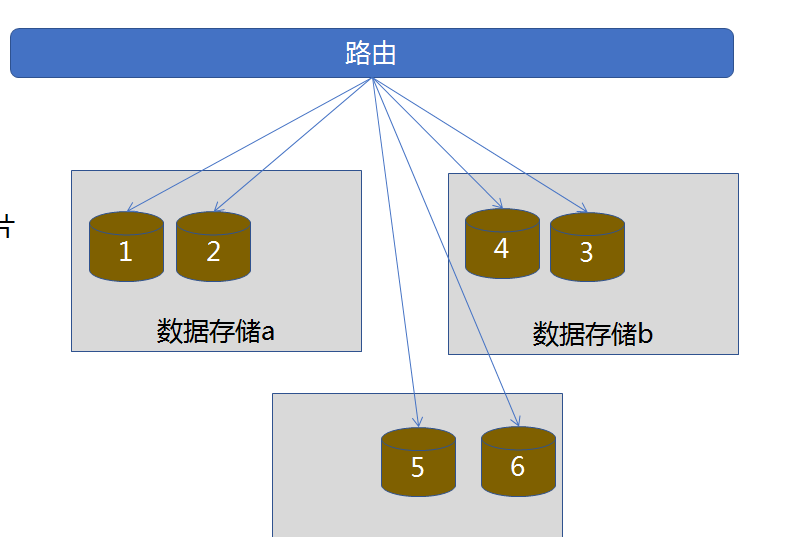
2、spring-session插件，实现session共享。

十五、空间换时间的方案

有一个模式：jdk有一个future模式，

十六、大数据量的切片方案

数据分片：数据的路由保持不变



1、设置数据片数量，比如100个柱

2、当前库，均分这100个柱

3、当加库扩容时，这100个柱动态迁移

100万的用户，

张三，数据查询时，------redis集群，----此时路由层知道张三的数据存在哪个柱上。

----- 数据和柱的对应关系，扩容时，此关系不变

1、微服务里的数据库，是私有库，对外不开放的。

2、微服务与分布式，是相辅相成的，不互拆。相互之间rpc调用，是无障碍的。