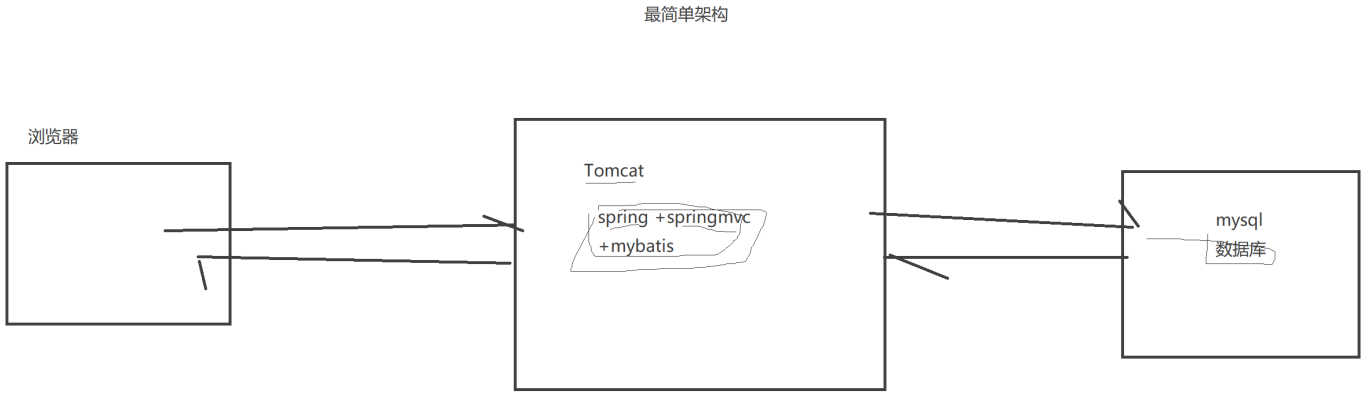
架 构 演 进

原创

xue_yun_xiang 于 2021-06-15 22:33:38 发布 116 收藏 6

分类专栏： JavaEE 文章标签： java 分布式 微服务架构

> java 工程师 常使用ssm（spring + springmvc + mybatis）开发后台应用



> 架构演进：作用就是为了给用户提供更好的服务（应对越来越多的用户，使用该服务造成的高并发，高可用问题）

一、开发环境&生产环境

开发环境

> 就是程序员开发调试代码的环境，在这个环境中都是测试数据，对用户没有任何影响，主要是为了便于调试 测试准备，数据可以随意的修改，对于工程师来说权限很好

> 编码 ：window mac
> 部署：linux (tomcat mysql)

生产环境

> 生产环境就是线上环境，就是用户使用的服务运行的环境，在这个环境中，程序员在开发测试阶段不可以 随意的接入，包括断点调试，修改线上的数据（mysql），也不能压力测试都不可以使用，只能在测试/开发环境使用

> 生产环境所有的数据都是用户产生/或者针对用户服务的数据，非常的重要，决不允许随意修改。

测试环境

> 和开发环境类似，主要用户与测试小组测试 和 压测

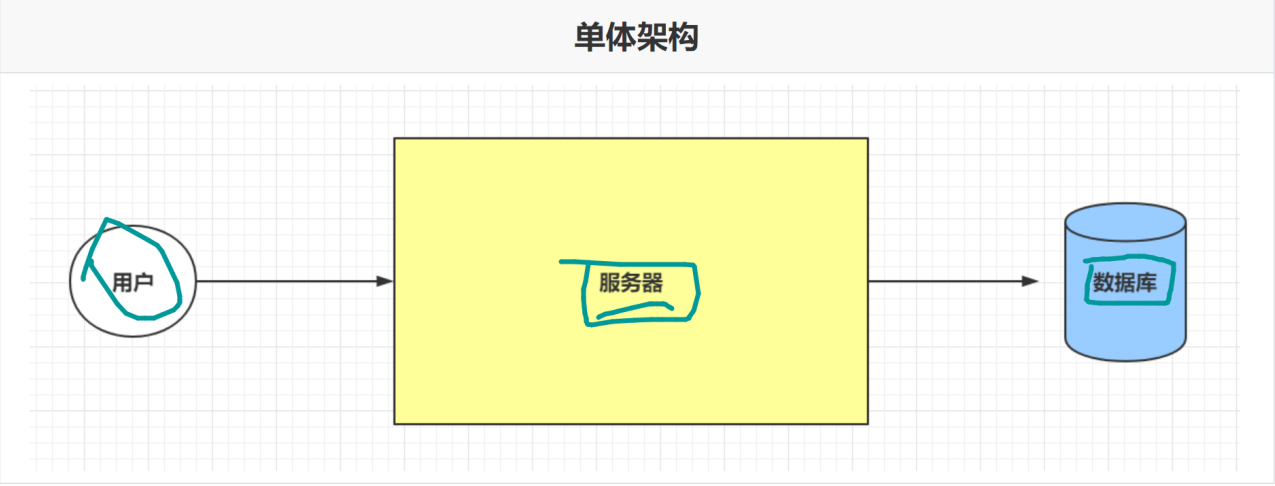
二、web 1.0& web2.0

单体架构

> 特点：用户少，内容少 对高可靠，高并发的性能要求都很低

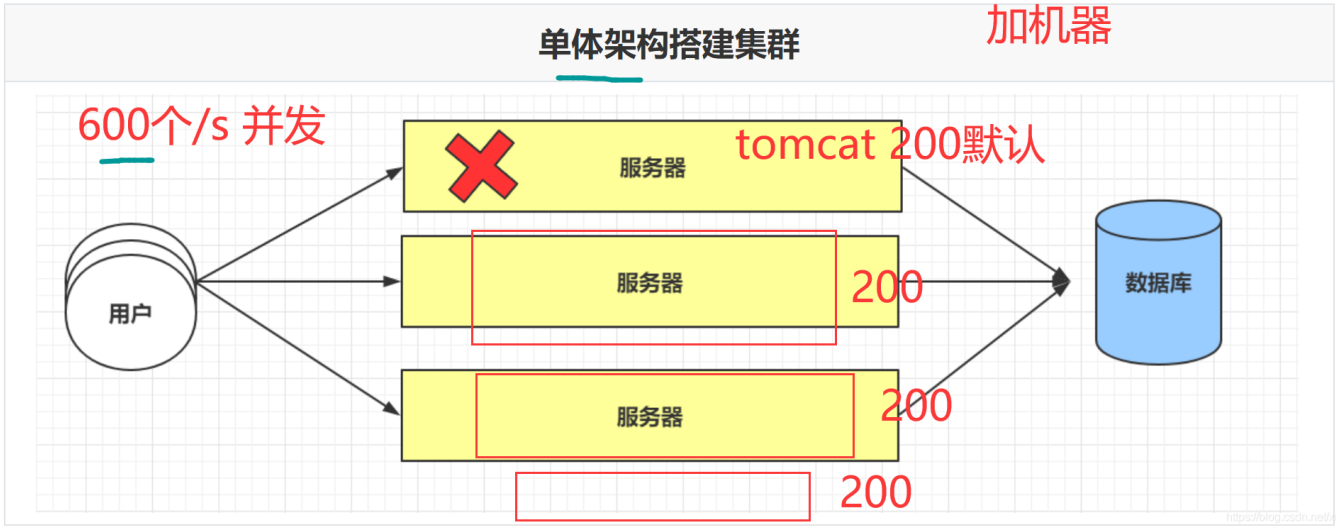
2.1 Web1.0阶段

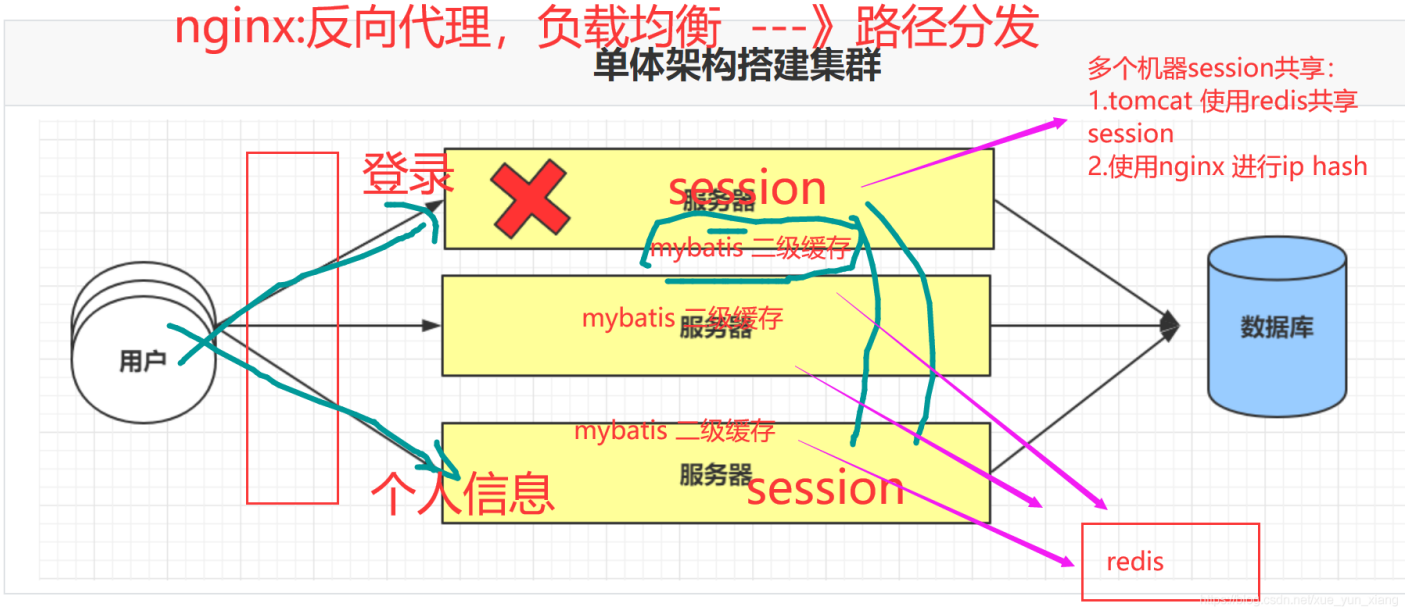
> 在Web1.0阶段，由于带宽不足，这时的项目大多是内容少，用户量也不多，甚至有一些项目不需要对外开放，对安全性和稳定性的要求是不高的。
>
> 单体架构就足以应对。



单体集群架构

1.路径分发问题（用户到底请求那一台服务器）
2.共享session问题
使用nginx ip_hash 解决
使用redis 共享session
3.多个服务器共享 mybatis 二级缓存为题
一定程度上可以解决用户高并发，高可用问题





三、垂直架构

在一个应用中：
登录、注册、商品模块、购物车、订单模块、个人中心、推荐模块、收藏、积分模块、支付模块

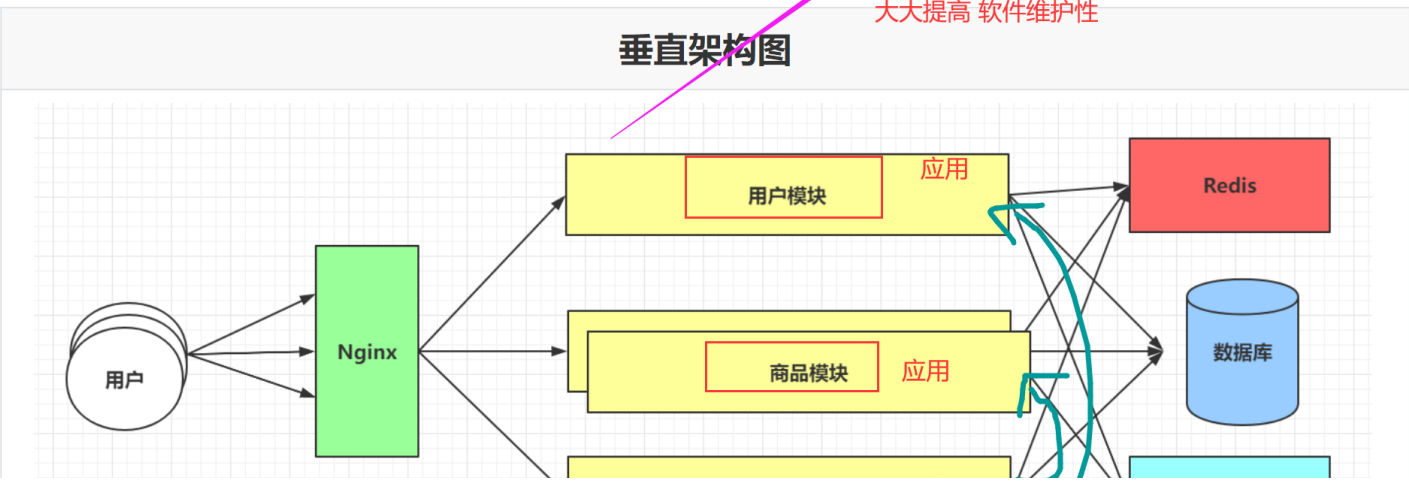
假如我们只需要对订单模块增加一个功能（修改一个bug）,所有的tomcat 服务器都要更新代码
出现问题：
1.所有的tomcat 都要更新代码
2.代码的维护性非常差（成本高，每个人都要熟悉各个模块）

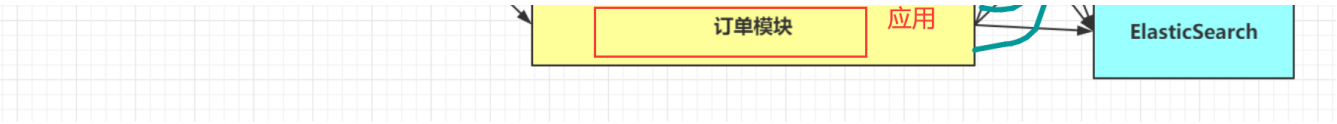
关于单体架构中，完美的体现了低内聚，高耦合，避开了开发的准则。

为了解决上述的各种问题，演进出了垂直架构。

高内聚，低耦合

大大提高 软件维护性



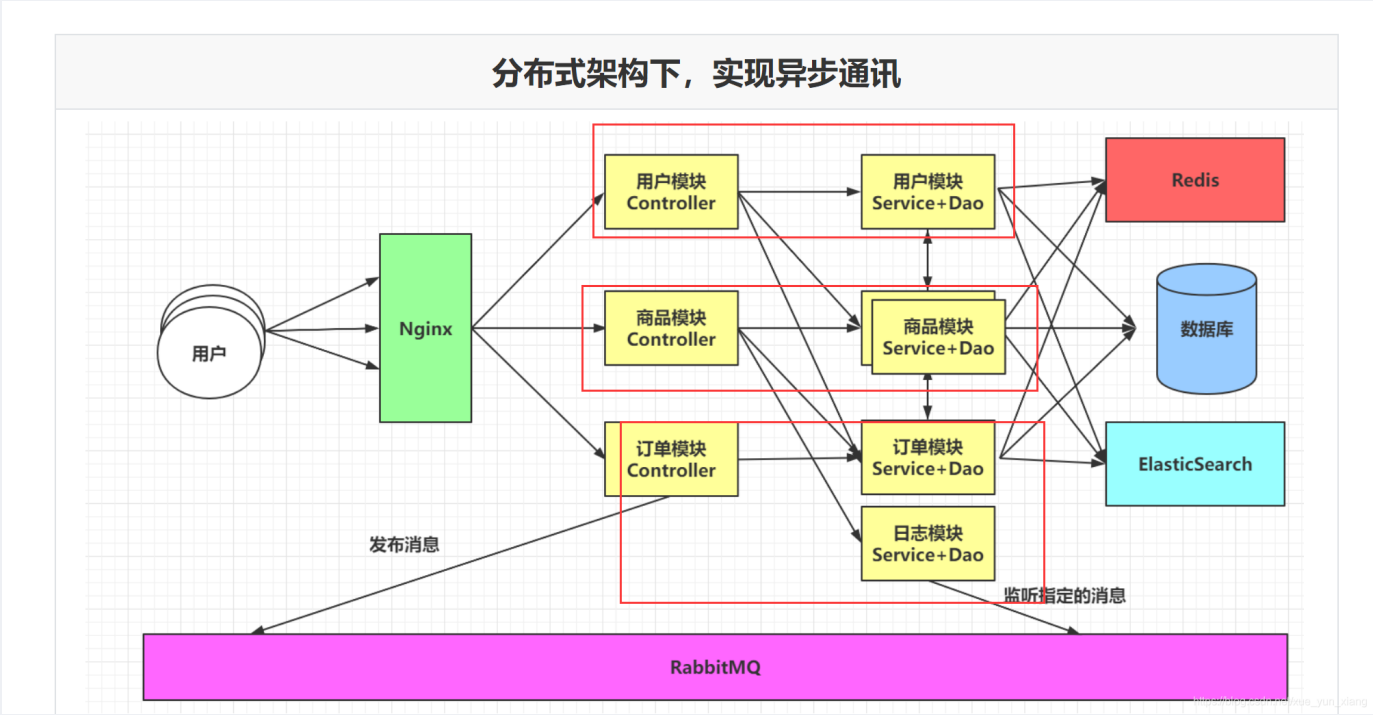


垂直架构可以解决代码维护性差的问题，有带来了一个最大的问题，多个模块之间如何通信？

3.1 模块之间通讯

同步通讯

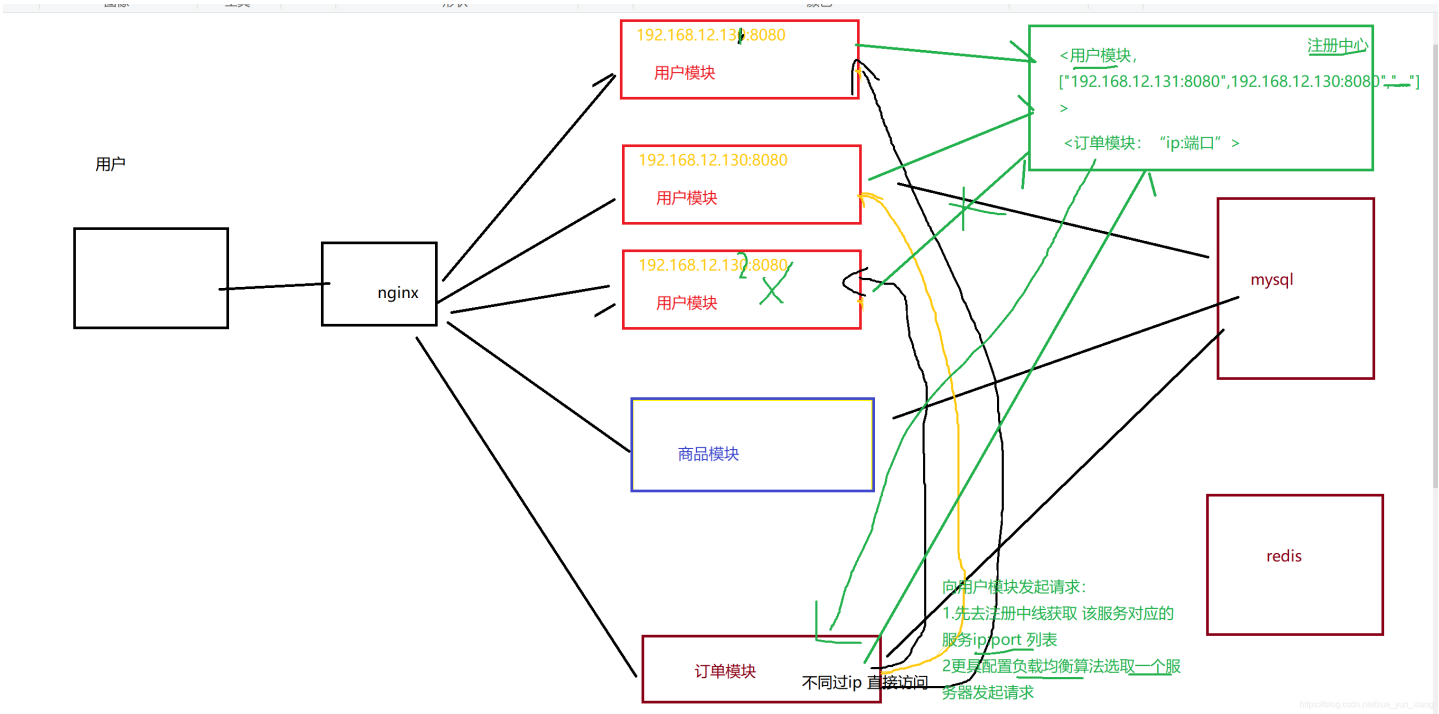
模块之间通过http 请求 发送请求数据，发起一个请求等着另外一个模块回复
实现：使用springcloud（底层http请求） 使用dubbo（rpc框架底层是 scoket 长连接（tcp）实现
相当于打电话
异步通讯
模块发出一个消息，就不用关心了，可以去干其他事情了，另外一个模块接收到消息，按照对应的业务处理就可以了
实现：使用消息队列实现（rabbitmq, rockermq,kafka ,redis 中的 list ）
相当于 模块件发送短息



3.2 服务之间通讯地址的维护

使用**注册中心**解决 各个服务模块之间如何快速的发现对应服务的ip 和端口，并且如果某一服务下线，发起调用的模块可以立即感知到，不会再次向 下线的服务ip 发起调用

解决方法：
nacos: 解决springcloud模块之间 注册发现
eureka:解决springcloud模块之间 注册发现 已经停止更新
zookeeper： 解决的dubbo 模块指的 注册发现
redis: 解决的dubbo 模块指的 注册发现



3.3 模块之间 请求的负载均衡

调用一个模块，如果该模块有多台机器提供服务，那么发起请求时就需要配置不同的负载均衡算法发起请求
负载均衡实现
Ribbon:解决的是springcloud 模块间请求的负载均衡
Feign：封装了 ribbon，更加简便 解决的是springcloud 模块间请求的负载均衡

3.4 服务降级/熔断

作用：避免 某一模块依赖其他模块，如果依赖的模块挂了，引起当前模块也出现问题
避免服务之间连锁雪崩的情况出现

当订单模块依赖用户模块，如果所有的用户都挂了，此时用户模块无法提供服务，而订单依赖用户模块，早程订单模块的不可用

如何解决？
此时我们就可以在订单模块中设置降级策略，如果所有用户模块关掉，降级策略生效（降级策略是返回默认的死数据）

淘宝首页-----》淘宝首页推荐商品模块----》推荐模块
假如推荐模块 挂了，那么淘宝首页商品模块 也无法提供服务，此时我们可以为淘宝首页设置降级策略，如果推荐模块挂了，我们就直接去 redis 拿 热门商品数据 作为首页推荐数据返回

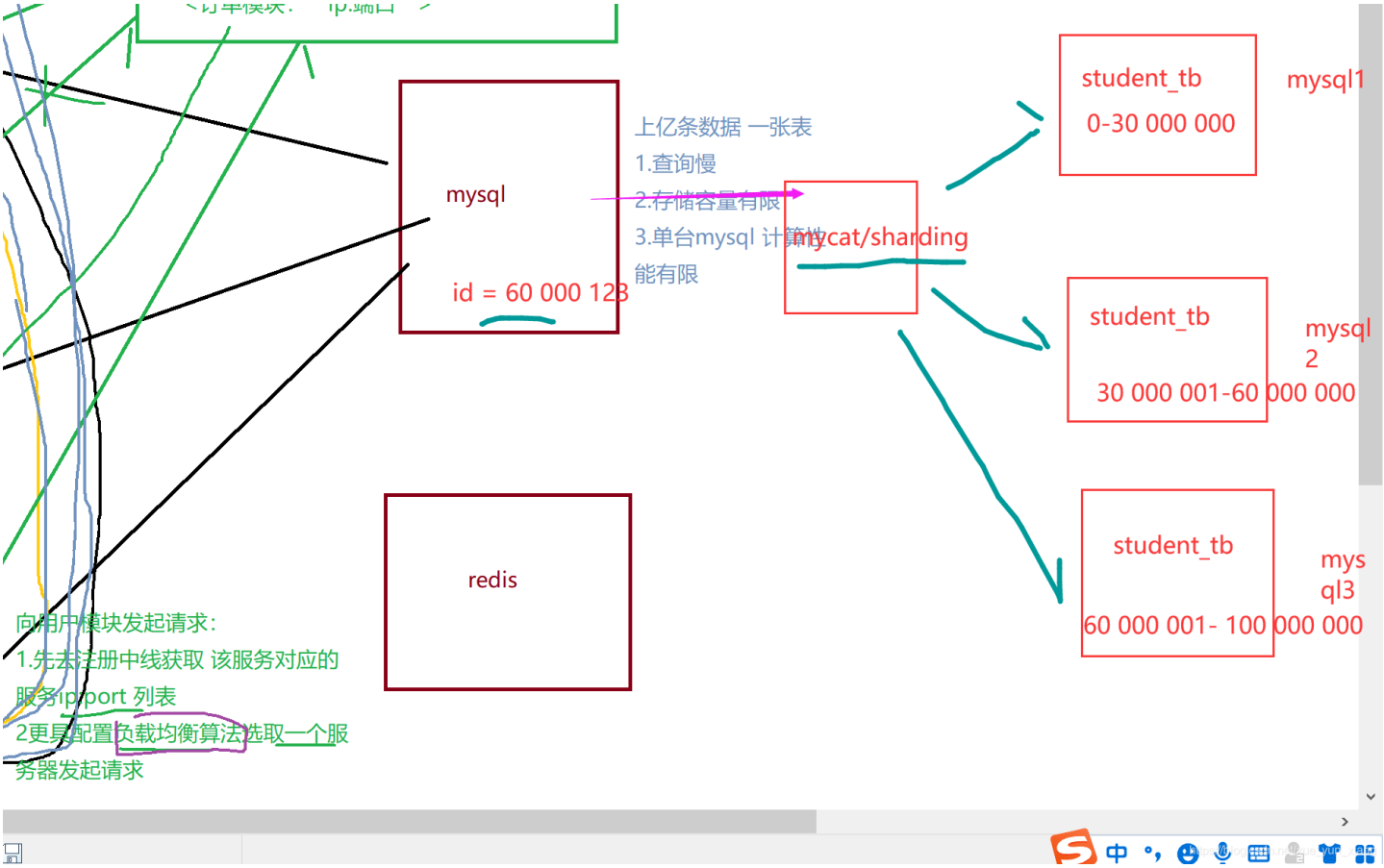
熔断策略：
当流量过大，超过阈值，当响应时间过长超过阈值，单位时间内异常数量超过阈值都会触发 熔断策略

> 实现：
> feign 配置降级
> hystrix 配置熔断 降级 springcloud
> sentinel（哨兵）配置熔断 降级策略 springcloud

3.5 海量的数据

海量数据的存储

mysql 数据表 最大是千万级别，如果数据量是上亿，必须使用mycate/sharding 进行分库分表



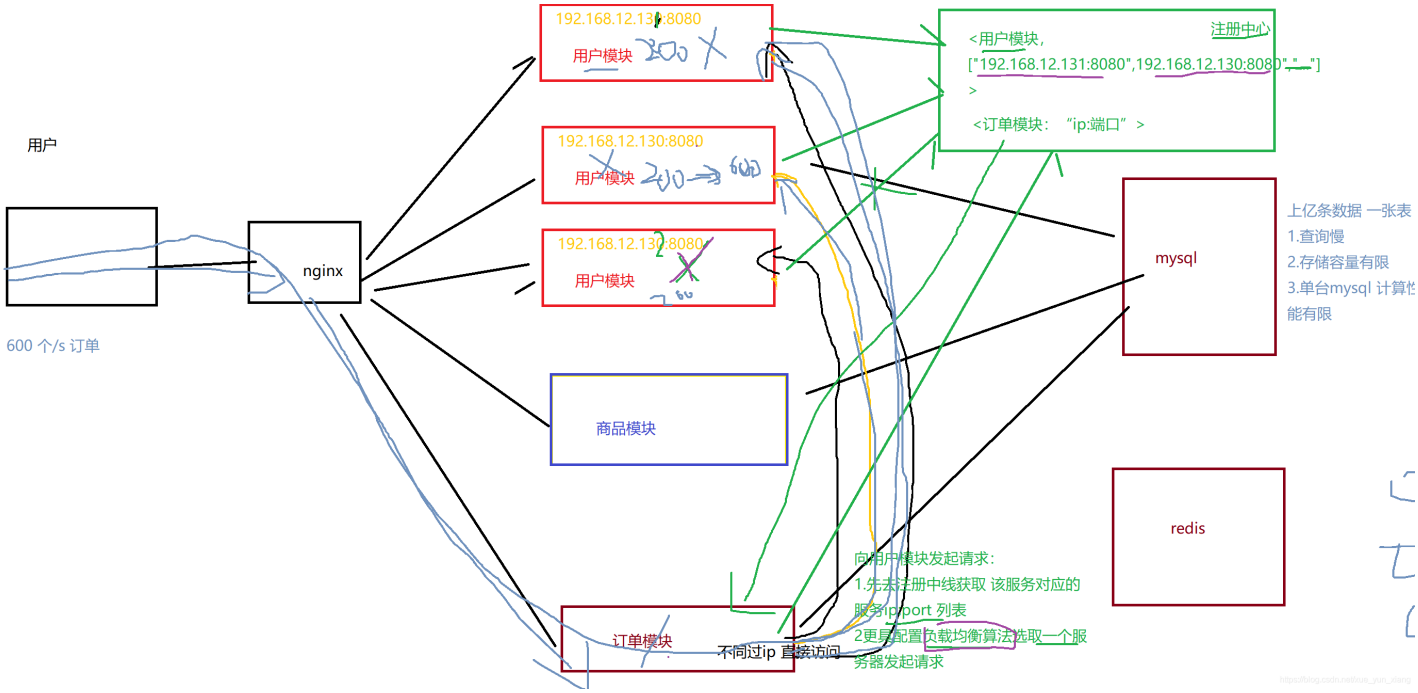
海量数据的查询/检索

> 可以使用es 搜索引擎解决

四、微服务架构

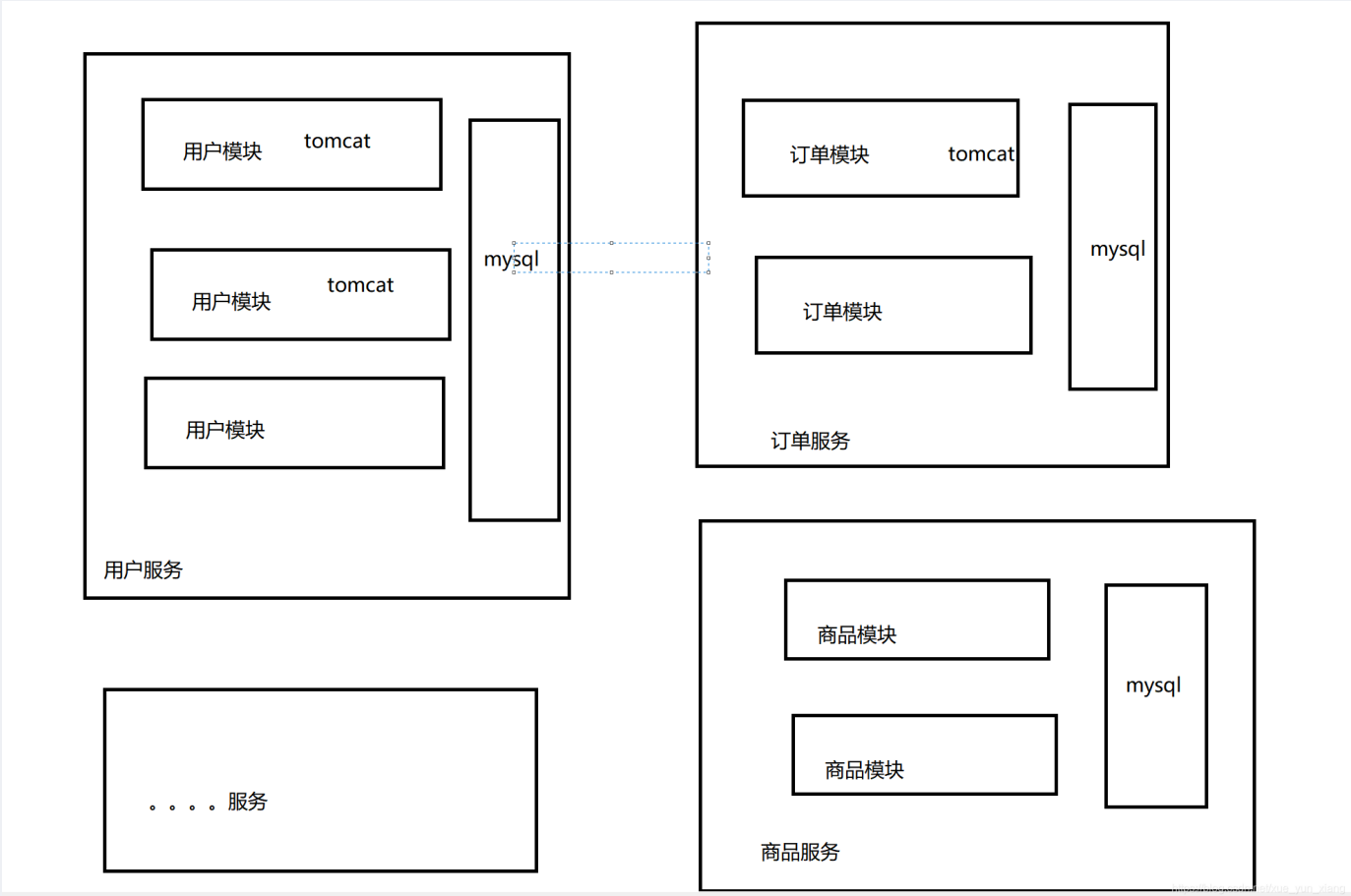
分布式架构

> 可以将一个应用拆分多个模块应用开发部署，还可以将一个数据库分为多个
> mysql 可以分库分表，redis 去中心化，多台tomcat可以通过nginx 负载均衡



微服务架构

微服务架构：微小的服务 微服务架构就是将一个应用 按照功能模块划分为多个独立应用，
每个应用可以单独的部署多台，这些应用可以通过http/tcp 互相调用，那么这种架构可以成为微服务架构



模块过多，使用docker 容器化部署

1.现在有很多tomcat 部署 不同的服务，服务器很多，如果人员介入很麻烦
2.各个服务的升级
3.服务中的tomcat 实例可以动态的扩容 和 缩减
使用 dockerimage(tomcat + ssm) +dokcker + swarm/k8s（可以动态增加某一容器的数量） + 云主机（阿里云）

分布式事务

生成订单
1.购物车模块
2.订单模块
3.库存模块
4.支付模块
在单应用中，可以让所有的模块共享一个sqlsession ,并且开启事务，来保证一致性
在微服务应用中，不可让所有的模块共享一个sql session ,使用单机开启事务 无法保证整体一致性

如何解决？
可以使用分布式事务解决 seata 框架解决

分布式锁

解决多个应用之间访问同一数据，保证线程安全（数据的一致性）
解决方案：
1使用 redis setnx 实现
2.使用zookeeper 临时有序节点实现