



哈尔滨工业大学 国家示范性软件学院

面向服务架构 淘宝技术架构演进

杨大易

2021/12/14

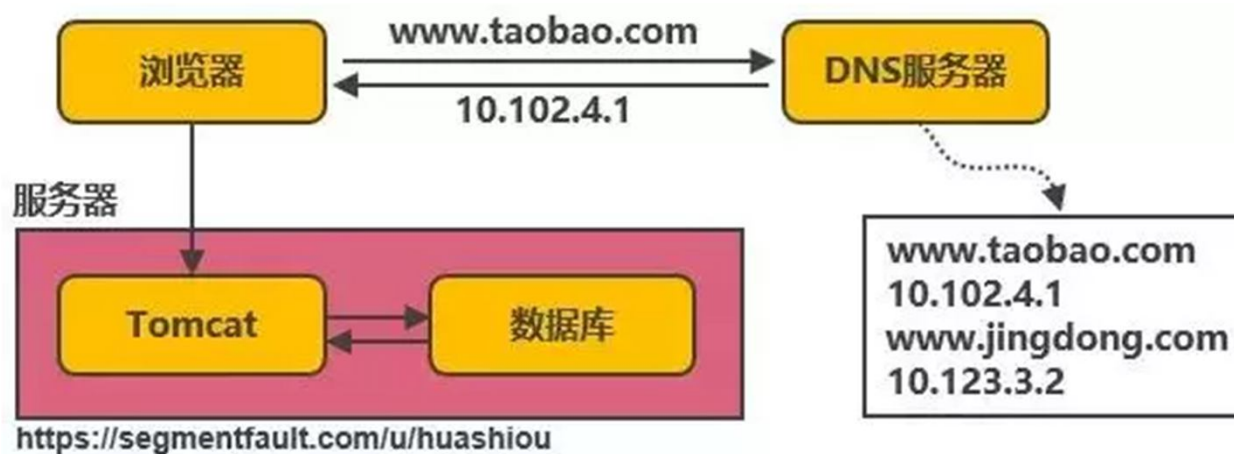


最初架构



❖ 单机架构

- 在网站最初期，应用数量与用户数量都较少，可以把Tomcat和数据库部署在同一台服务器上



第一次演进



❖ Tomcat与数据库分开部署

- Tomcat和数据库竞争资源，单机性能不足支撑业务。

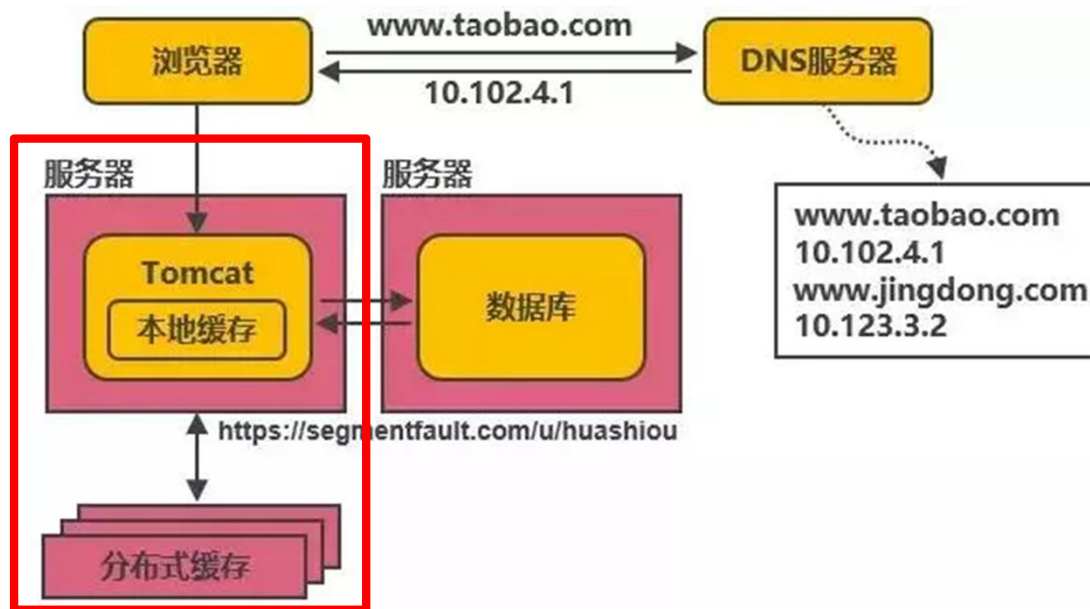


第二次演进



❖ 引入本地缓存和分布式缓存

- 并发的读写数据库成为瓶颈
- 使用memcached作为本地缓存，或使用Redis作为分布式缓存

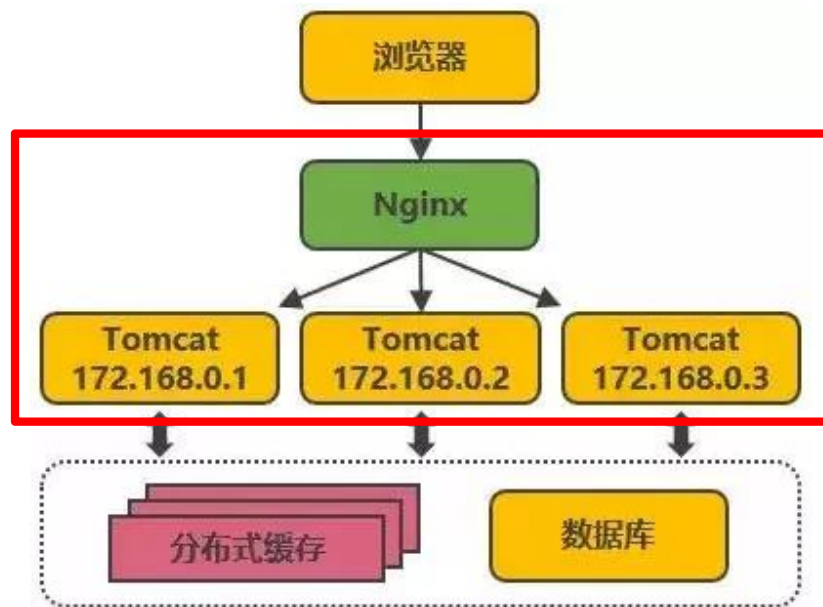


第三次演进



❖ 引入反向代理实现负载均衡

- 并发压力主要落在单机的Tomcat上，响应逐渐变慢
- 在多台服务器上分别部署Tomcat，使用反向代理软件（Nginx）把请求均匀分发到每个Tomcat 中

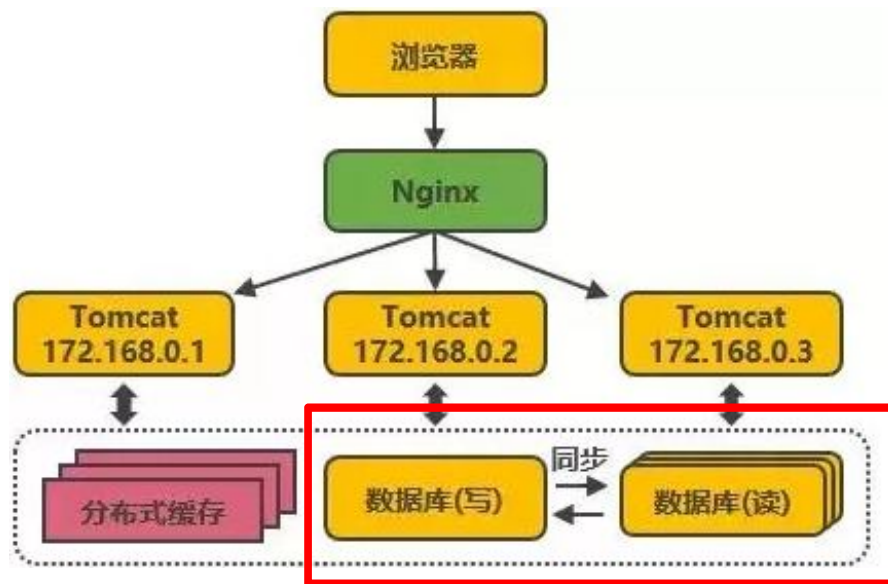


第四次演进



❖ 数据库读写分离

- 并发量的增长使单机的数据库成为新的瓶颈
- 把数据库划分为读库和写库，读库可以有多个，通过同步机制把写库的数据同步到读库

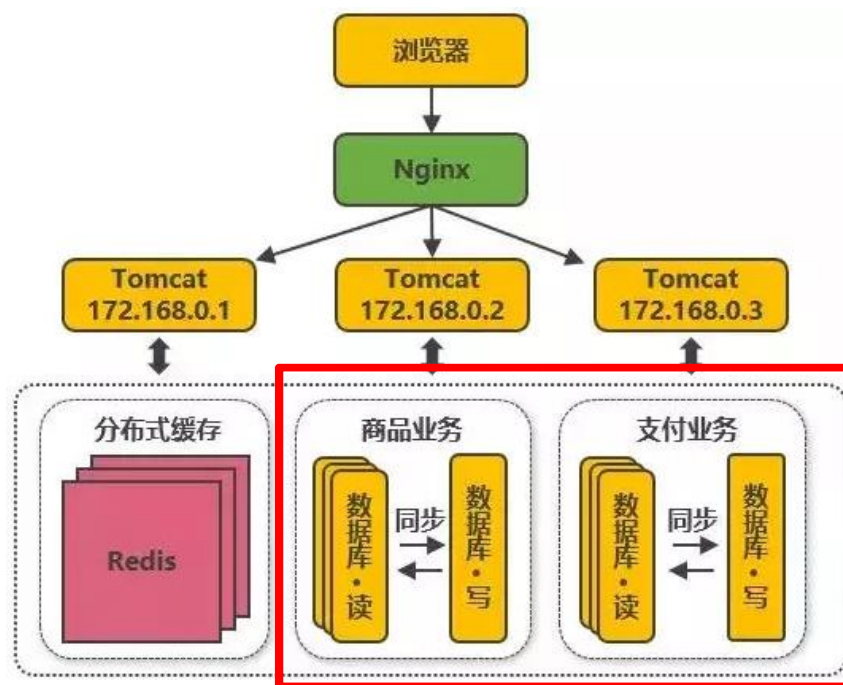


第五次演进



❖ 数据库按业务分库

- 不同业务之间的访问量差距较大，相互影响性能
- 对于访问量大的业务，可部署更多的服务器来支撑

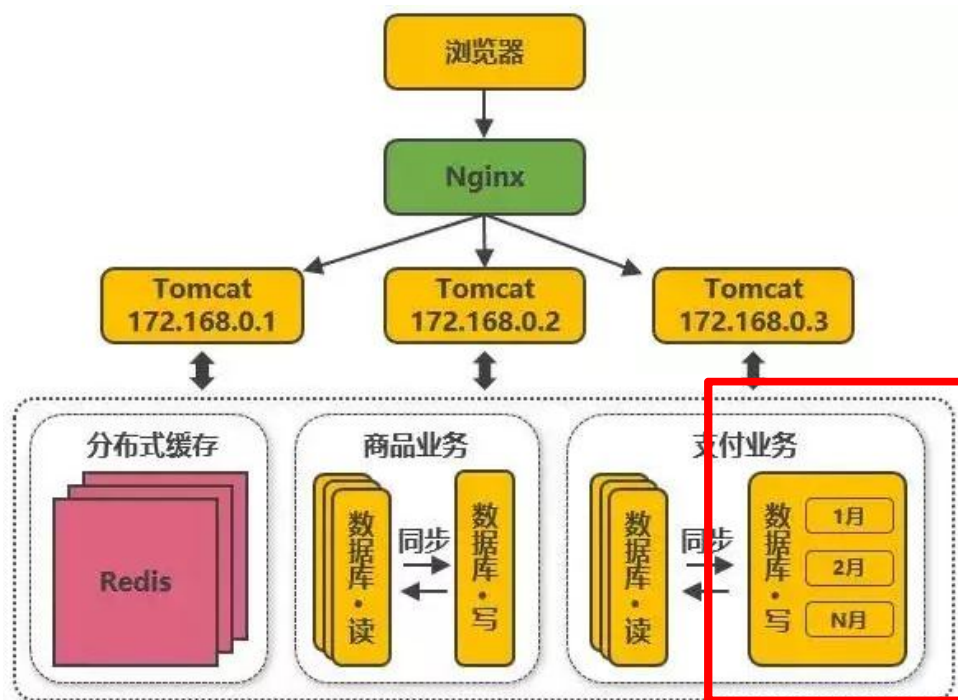


第六次演进



❖ 大表拆分为小表

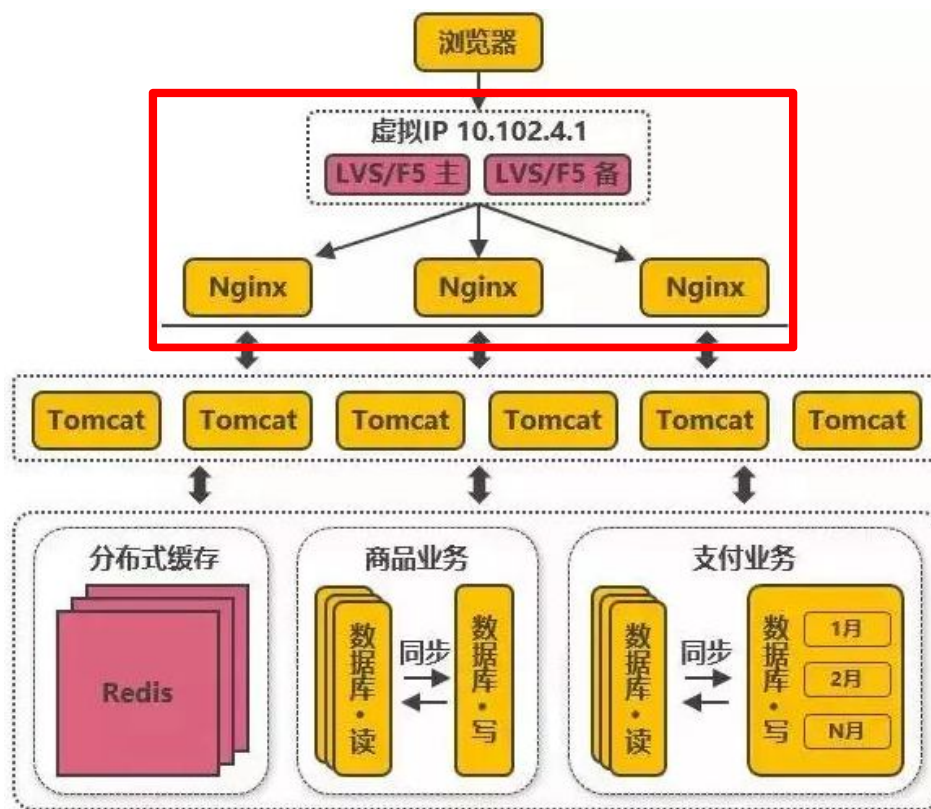
- 单机的写库逐渐达到性能瓶颈
- 将字段多、数据量大的数据表拆分为小表



第七次演进



- ❖ 使用LVS或F5来使多个Nginx负载均衡
 - 单机的Nginx成为瓶颈

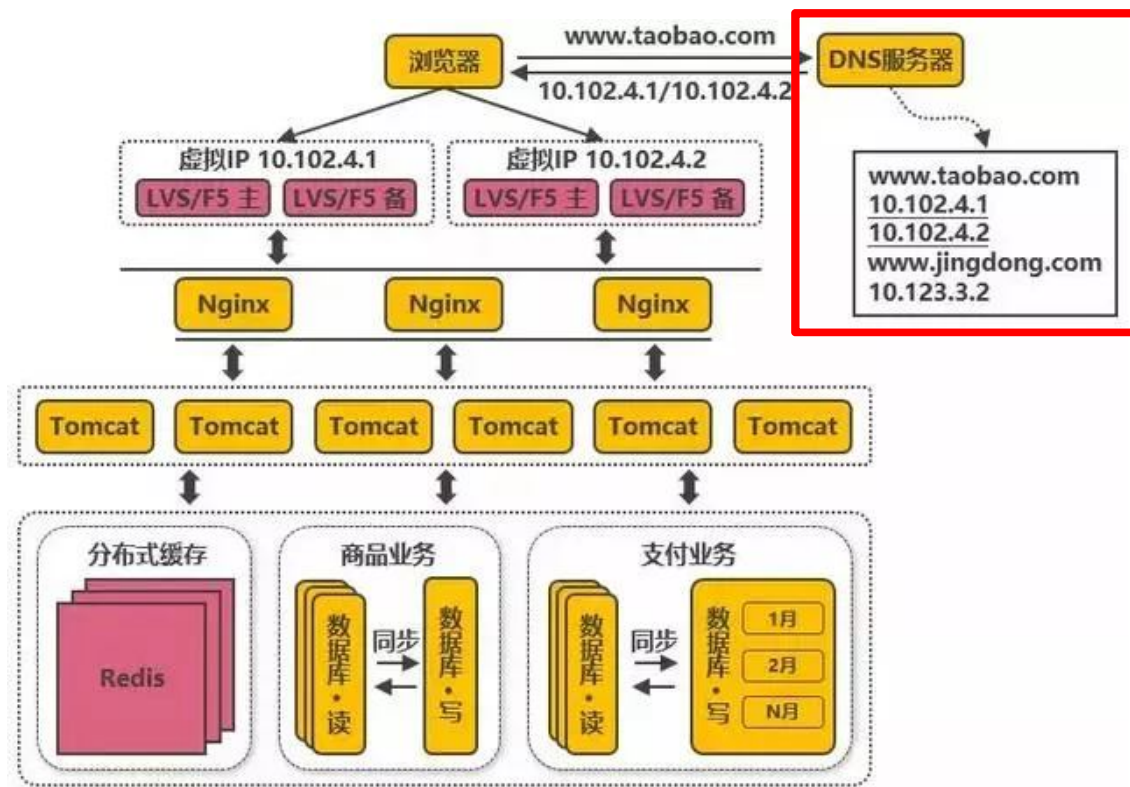


第八次演进



❖ 通过DNS轮询实现机房间的负载均衡

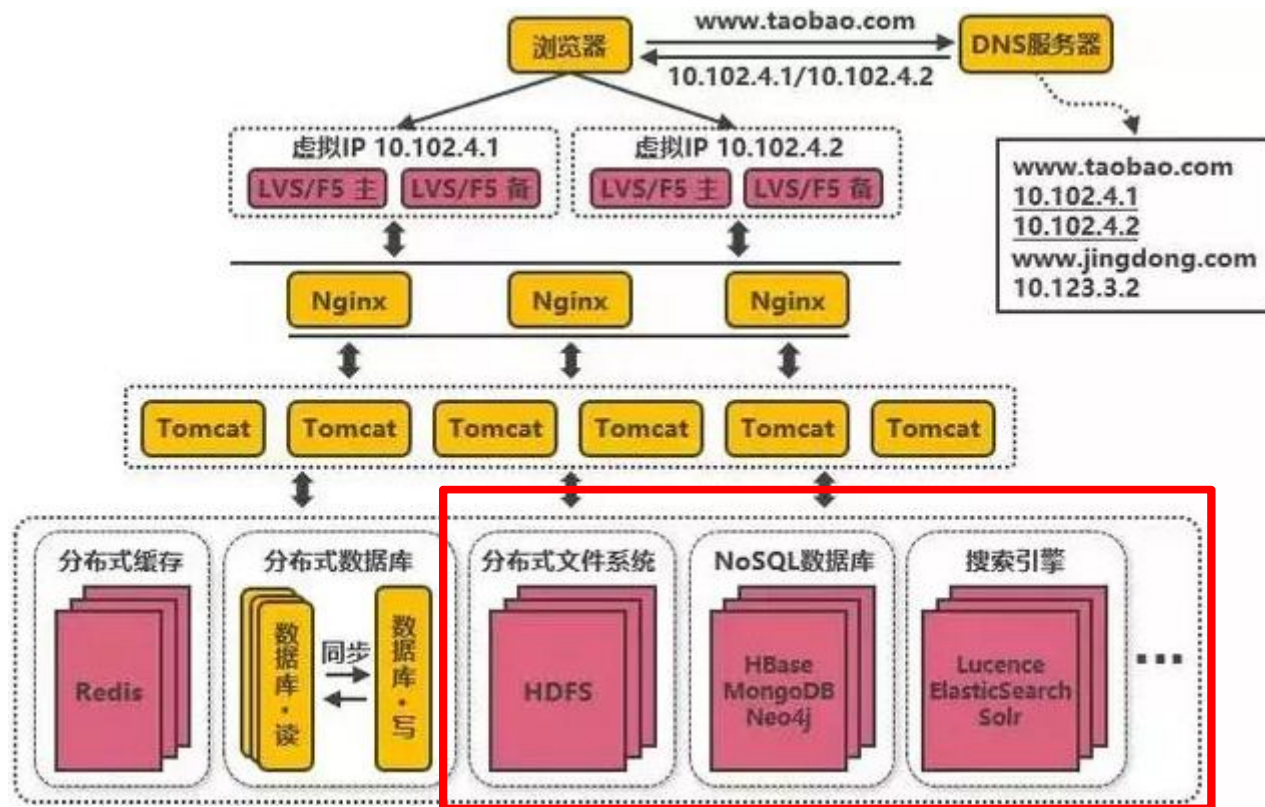
- LVS服务器达到瓶颈
- 在DNS服务器中可配置一个域名对应多个IP地址，每个IP地址对应到不同的机房里的虚拟IP



第九次演进



- ❖ 引入NoSQL数据库和搜索引擎等技术
 - 当数据多到一定规模时，数据库不再适用于复杂查询

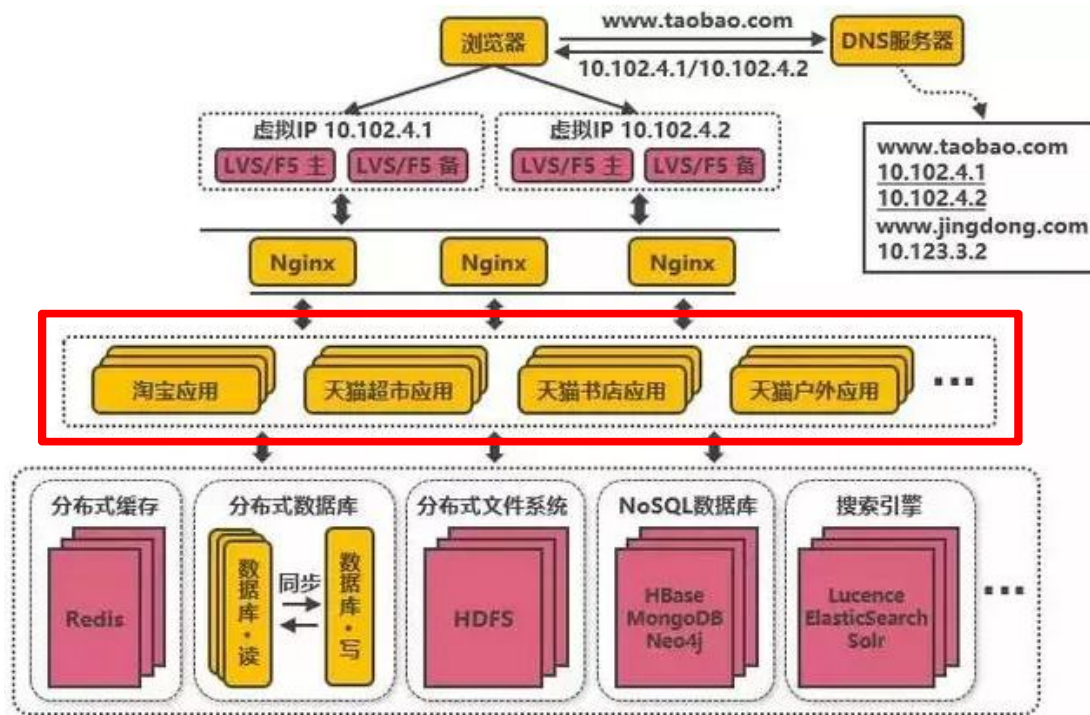


第十次演进



❖ 大应用拆分为小应用

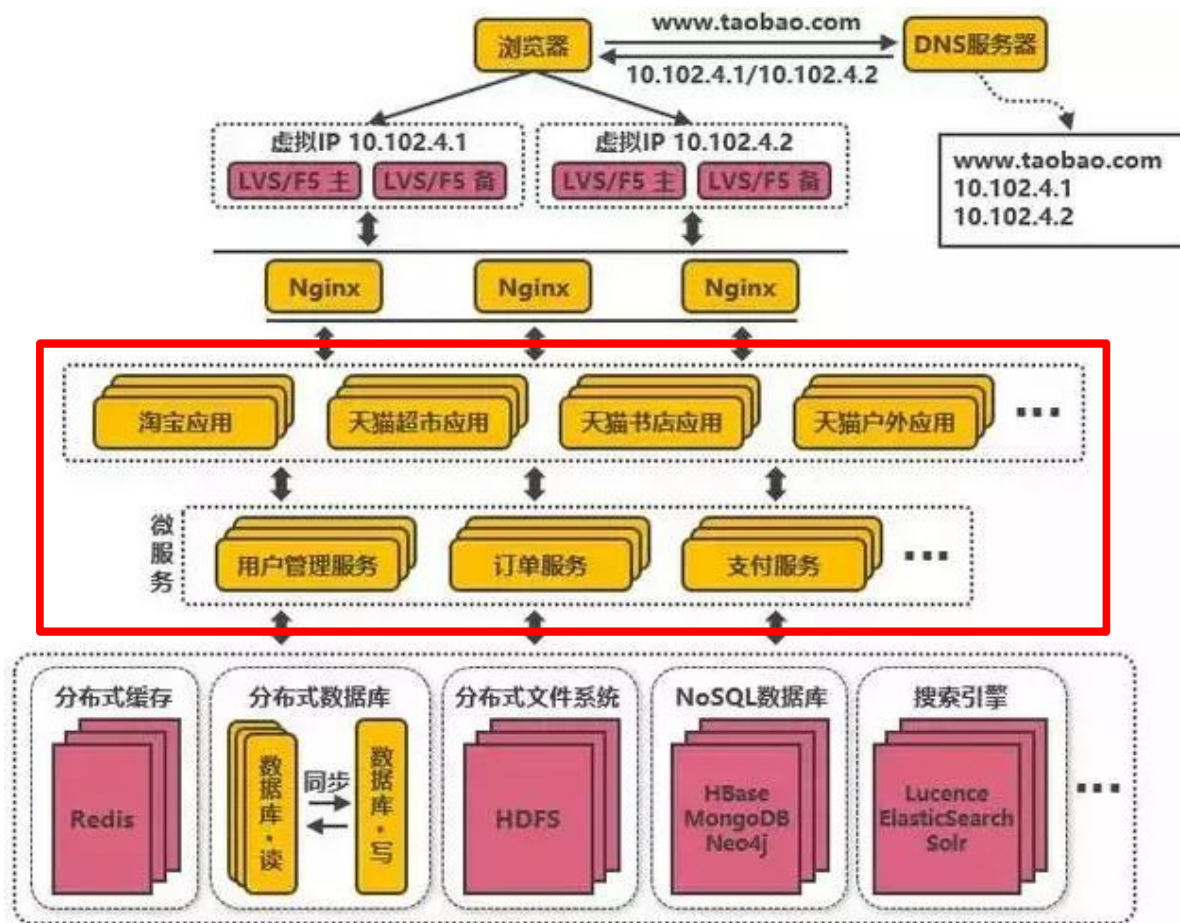
- 按照业务板块来划分应用代码，使单个应用的职责更清晰，相互之间可以做到独立升级迭代



第十一次演进



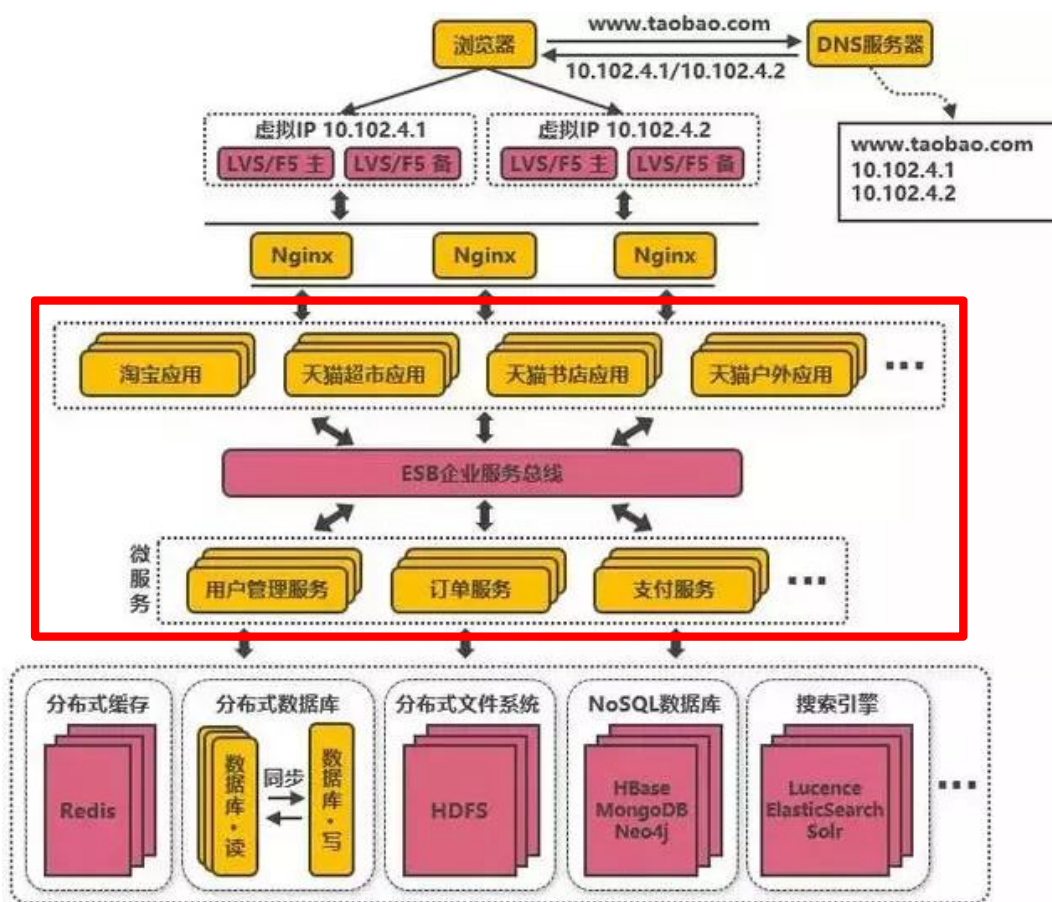
❖ 复用的功能抽离成微服务



第十二次演进



❖ 引入ESB屏蔽服务接口的访问差异

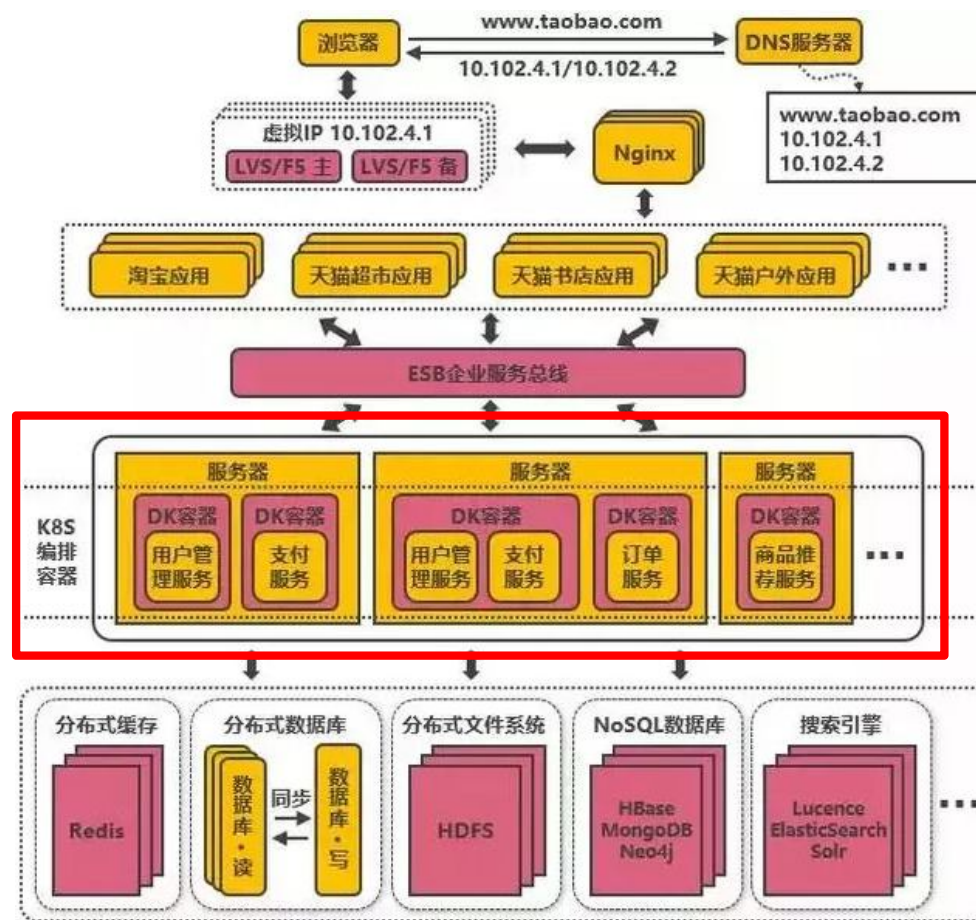


第十三次演进



❖ 容器化技术实现运行环境隔离与动态服务管理

- 服务部署、运维困难

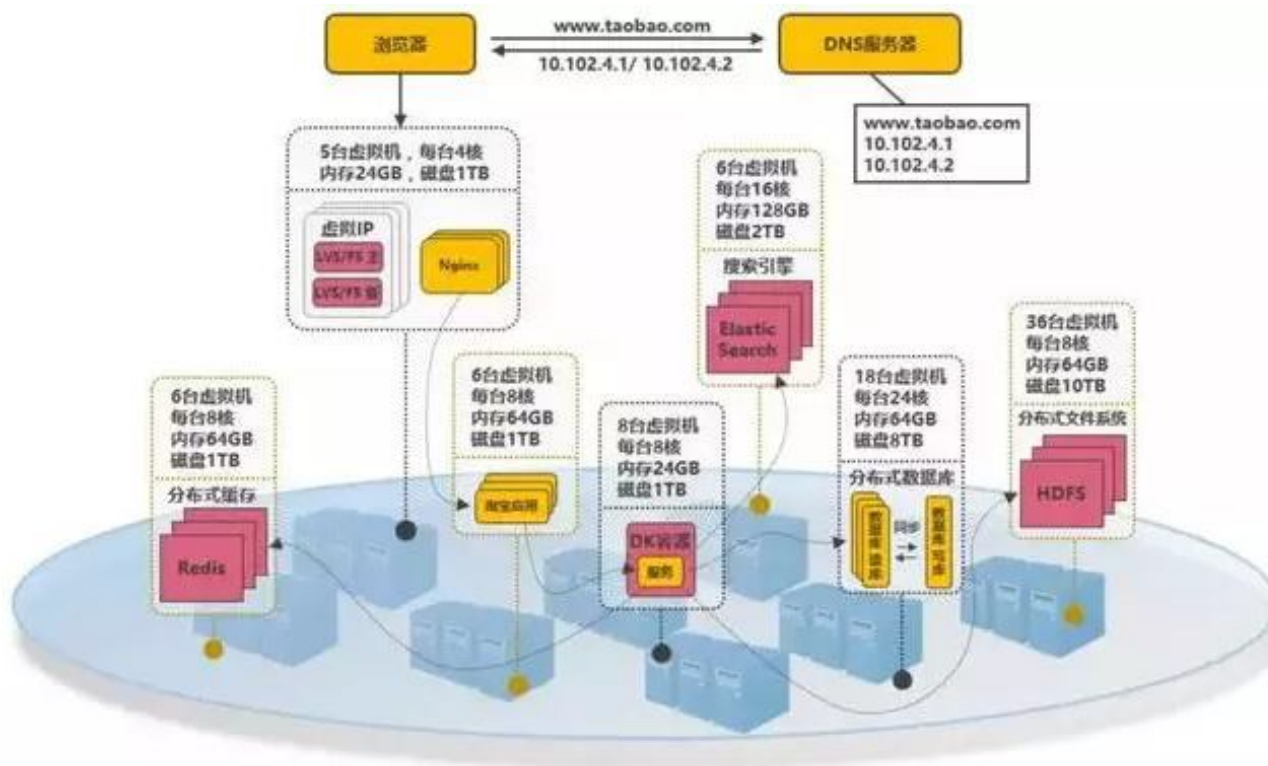


第十四次演进



❖ 以云平台承载系统

- 机器自身成本和运维成本都极高，资源利用率低



架构演进总结



- ❖ 成熟的架构方案不是一步搭建完成的，都是逐渐迭代演进而成。
- ❖ 不同业务系统演进路线并不完全一致。
- ❖ 分布式服务架构是构建大型应用软件系统的有效技术手段之一。





哈尔滨工业大学 国家示范性软件学院

谢谢！