

吃透这套架构演化图，从零搭建Web网站也不难！

张志朋

2017-10-27 11:04:49

455

作者介绍

张志朋，资深Java爱好者，深耕于在线教育领域，个人博客：<https://blog.52itstyle.com> (<https://blog.52itstyle.com>)。

前言

工作也有几多年了，无论是身边遇到的还是耳间闻到的，多多少少也积攒了自己的一些经验和思考，当然，博主并没有太多接触高大上的分布式架构实践，相对比较零碎，随时补充。

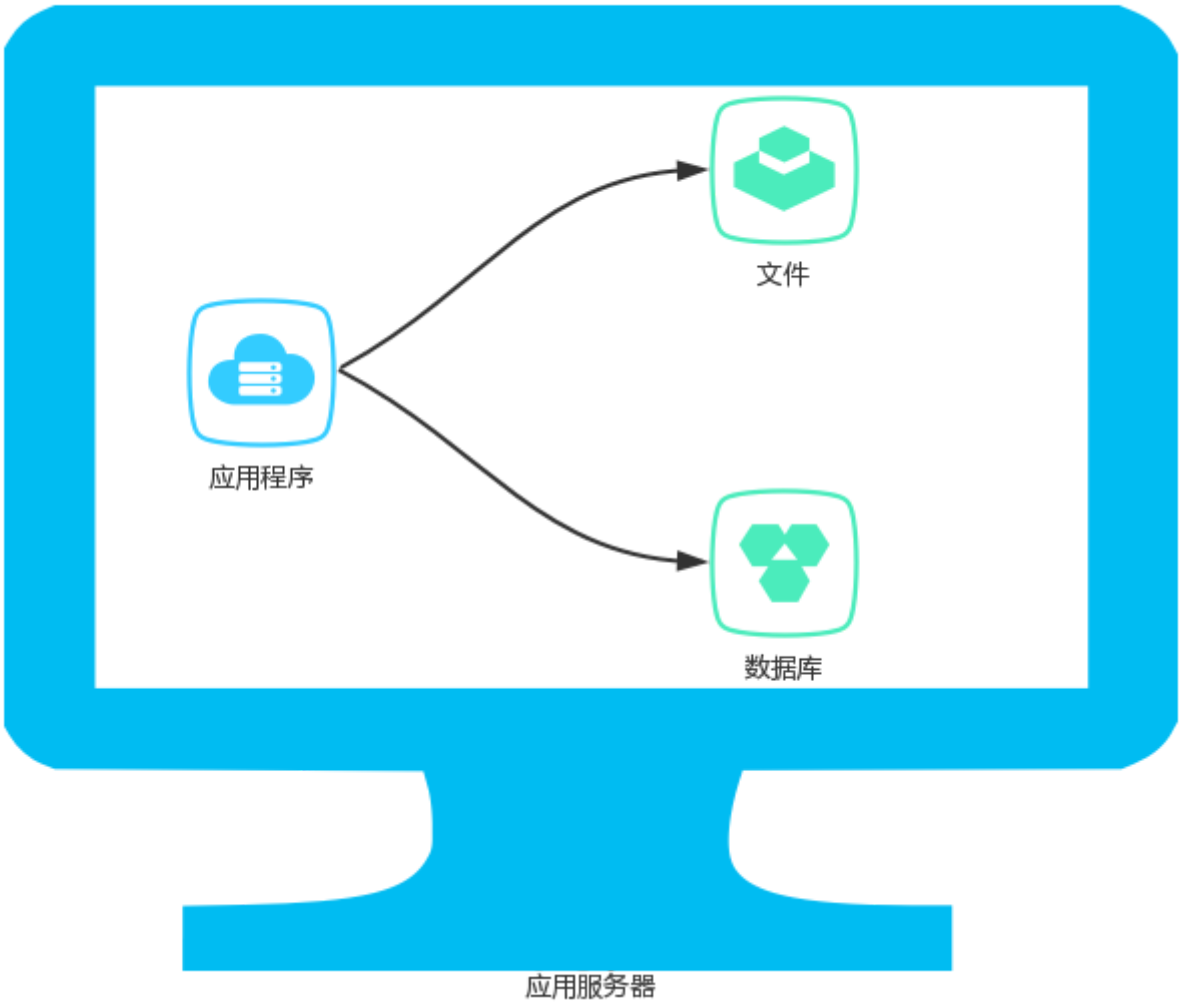
俗话说得好，冰冻三尺非一日之寒，滴水穿石非一日之功，罗马也不是一天就建成的，当然对于我们开发人员来说，一个好的架构也不是一蹴而就的。

初始搭建

开始的开始，就是各种框架一搭，然后扔到Tomcat容器中跑就是了，这时候我们的文件、数据库、应用都在一个服务器上。



(http://w

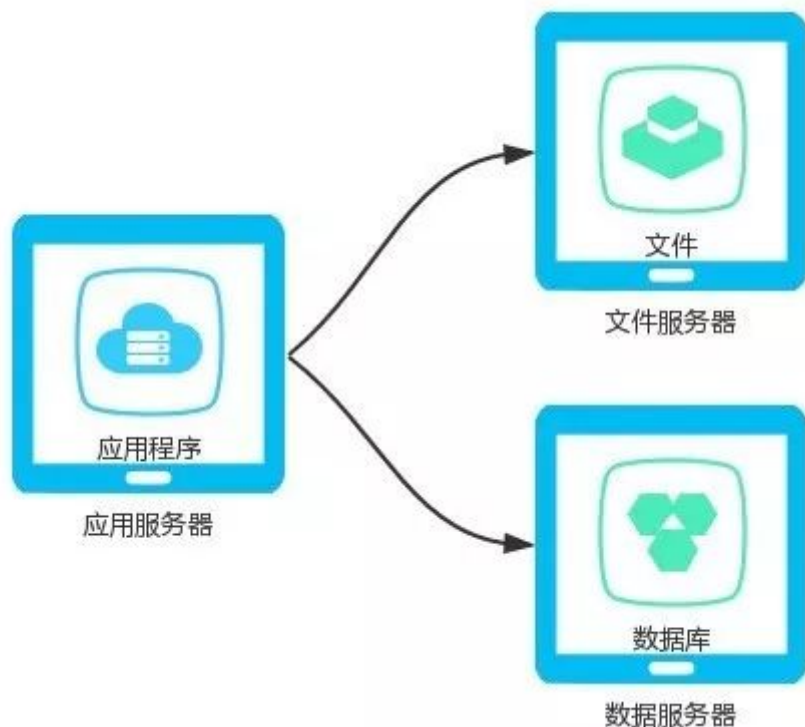


服务分离

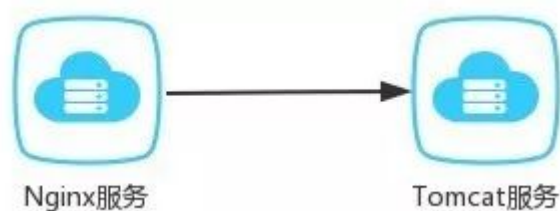
随着系统的的上线，用户量也会逐步上升，很明显一台服务器已经满足不了系统的负载，这时我们就要在服务器还没有超载时，提前做好准备。

由于我们是单体架构，优化架构在短时间内是不现实的，增加机器是一个不错的选择。这时，我们可能要把应用和数据库服务单独部署，如果有条件也可以把文件服务器单独部署。

(http://w



反向代理



为了提升服务处理能力，我们在Tomcat容器前加一个代理服务器，我一般使用Nginx，当然你如果更熟悉Apache也未尝不可。

用户的请求发送给反向代理，然后反向代理把请求转发到后端的服务器。

严格意义上来说，Nginx是属于Web服务器，一般处理静态HTML、CSS、JS请求，而Tomcat属于Web容器，专门处理JSP请求，当然Tomcat也是支持html的，只是效果没Nginx好而已。

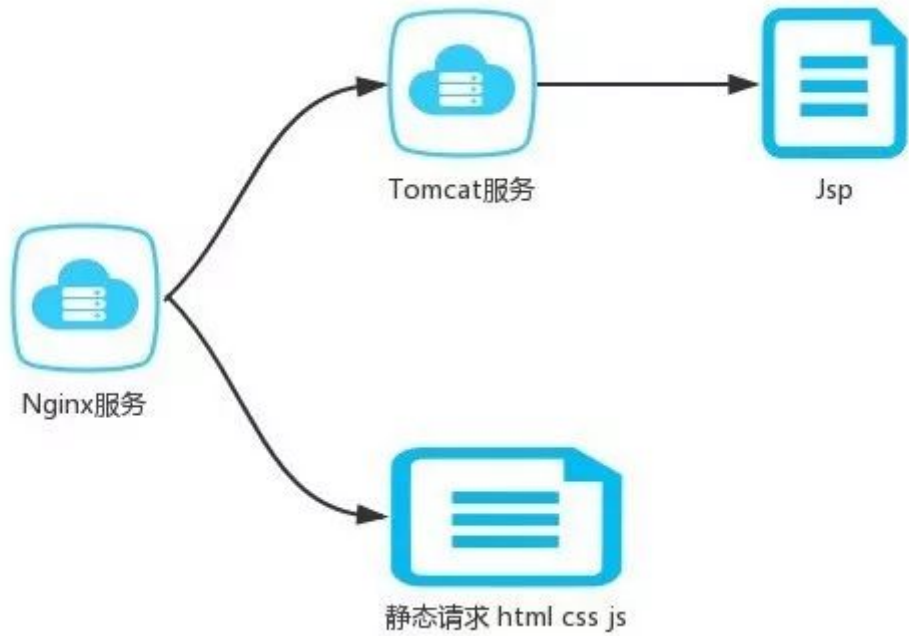
反向代理的优势，如下：

- 隐藏真实后端服务
- 负载均衡集群
- 高可用集群
- 缓存静态内容实现动静分离
- 安全限流
- 静态文件压缩
- 解决多个服务跨域问题
- 合并静态请求(HTTP/2.0后已经被弱化)

(http://w

- 防火墙
- SSL以及http2

动静分离



基于以上Nginx反向代理，我们还可以实现动静分离，静态请求如HTML、CSS、JS等请求交给Nginx处理，动态请求分发给后端Tomcat处理。

Nginx 升级到1.9.5+可以开启HTTP/2.0时代，加速网站访问。

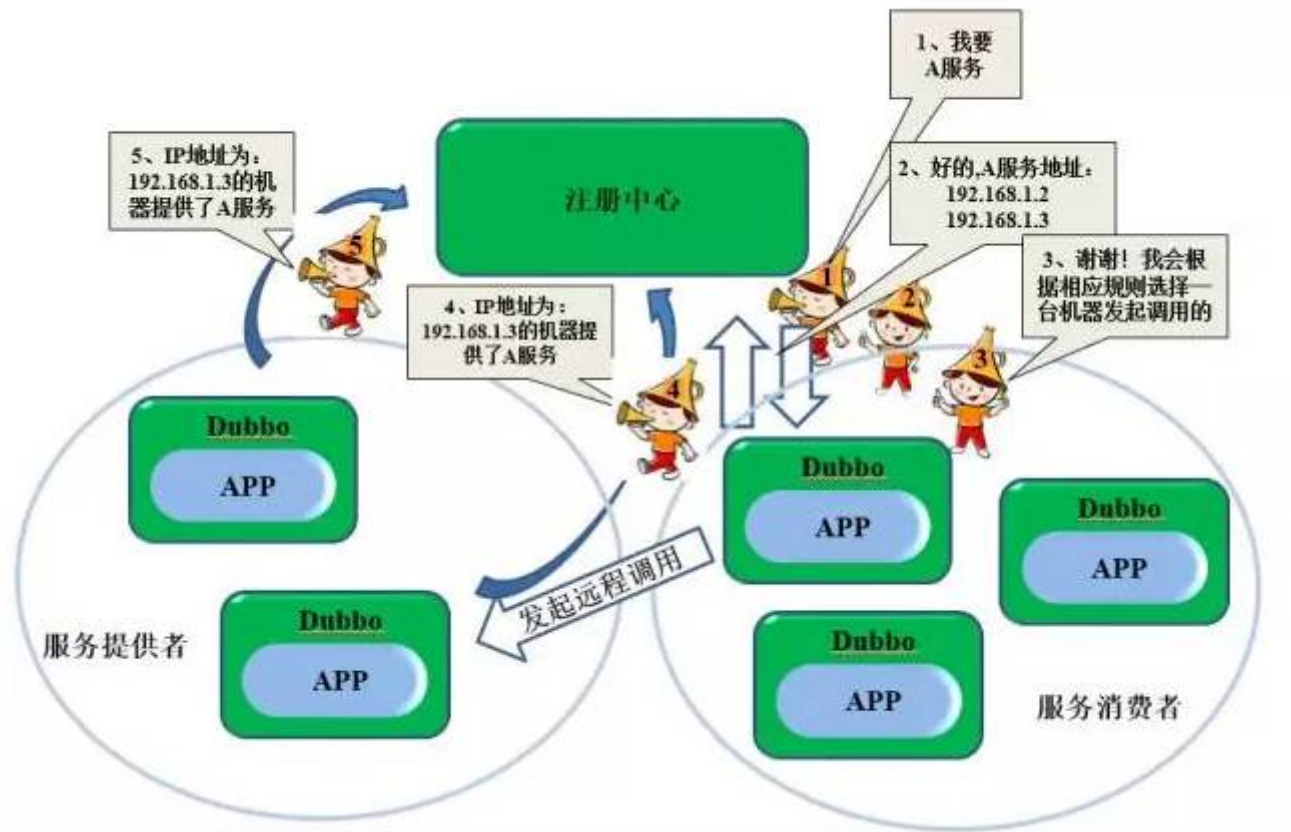
当然，如果公司不差钱，CDN也是一个不错的选择。

服务拆分

在这分布式微服务已经普遍流行的年代，其实我们没必要踩过多的坑，就很容易进行拆分。市面上已经有相对比较成熟的技术，比如阿里开源的Dubbo(官方明确表示已经开始维护了)，Spring家族的Spring Cloud，当然具体如何去实施，无论是技术还是业务方面都要有很好的把控。

Dubbo

(http://w



SpringCloud

- 服务发现——Netflix Eureka
- 客服端负载均衡——Netflix Ribbon
- 断路器——Netflix Hystrix
- 服务网关——Netflix Zuul
- 分布式配置——Spring Cloud Config

微服务与轻量级通信

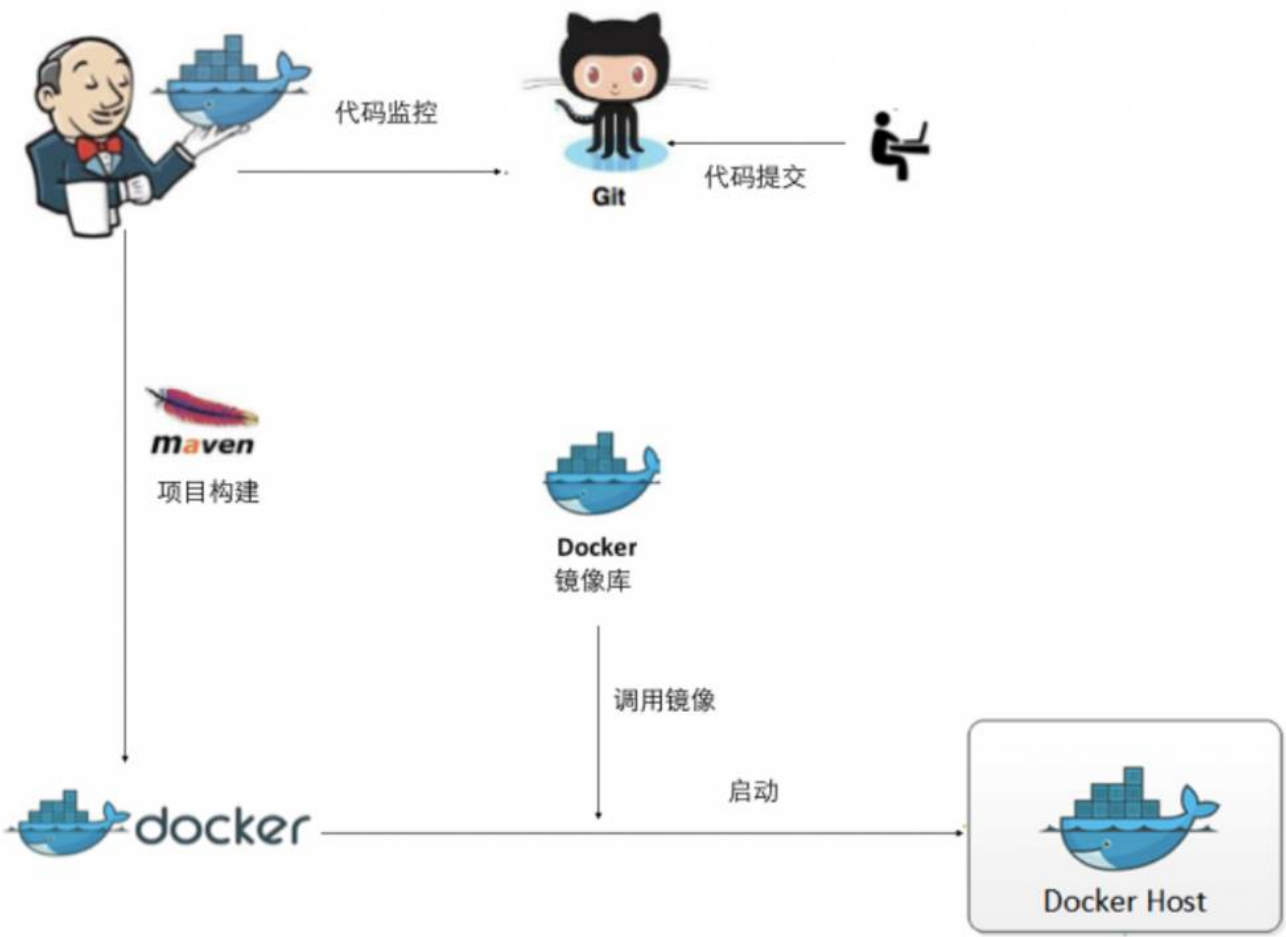
- 同步通信和异步通信
- 远程调用RPC
- REST
- 消息队列

持续集成部署

服务拆分以后，随着而来的就是持续集成部署，你可能会用到以下工具：Docker、Jenkins、Git、Maven。

图片源于网络，基本拓扑结构如下所示：

(http://w

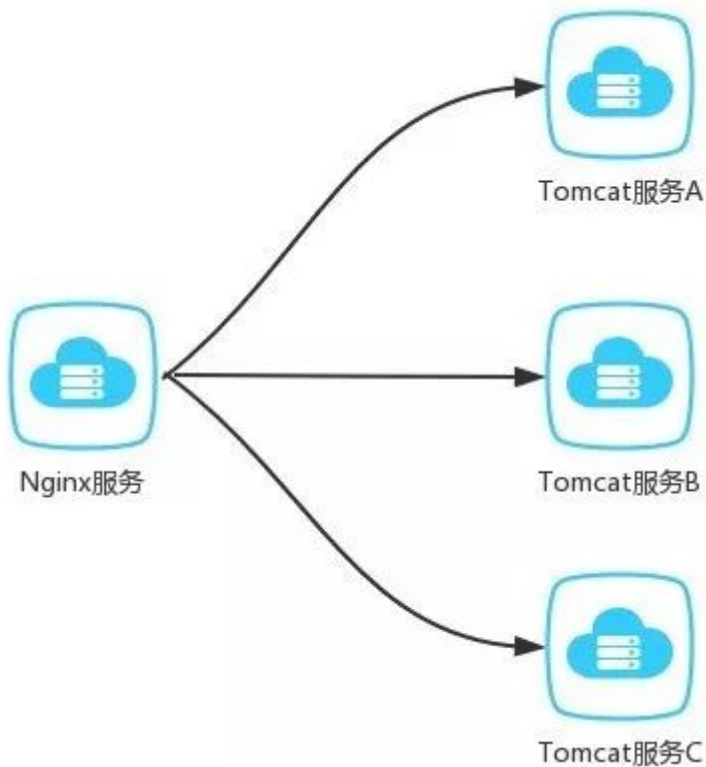


整个持续集成平台架构演进到如下图所示：



服务集群

Linux集群主要分成三大类(高可用集群， 负载均衡集群， 科学计算集群)。其实，我们最常见的也是生产中最常接触到的就是负载均衡集群。



负载均衡实现

- DNS负载均衡，一般域名注册商的dns服务器不支持，但博主用的阿里云解析已经支持
- 四层负载均衡(F5、LVS)，工作在TCP协议下
- 七层负载均衡(Nginx、haproxy)，工作在Http协议下

分布式session

大家都知道，服务一般分为有状态和无状态，而分布式session就是针对有状态的服务。

分布式Session的几种实现方式

- 基于数据库的Session共享
- 基于resin/tomcat web容器本身的session复制机制
- 基于oscache/Redis/memcached 进行 session 共享。
- 基于cookie 进行session共享

分布式Session的几种管理方式

Session Replication 方式管理 (即session复制)

(http://w

- 简介：将一台机器上的Session数据广播复制到集群中其余机器上
- 使用场景：机器较少，网络流量较小
- 优点：实现简单、配置较少、当网络中有机器Down掉时不影响用户访问
- 缺点：广播式复制到其余机器有一定延时，带来一定网络开销

Session Sticky 方式管理

- 简介：即粘性Session、当用户访问集群中某台机器后，强制指定后续所有请求均落到此机器上
- 使用场景：机器数适中、对稳定性要求不是非常苛刻
- 优点：实现简单、配置方便、没有额外网络开销
- 缺点：网络中有机器Down掉时、用户Session会丢失、容易造成单点故障

缓存集中式管理

- 简介：将Session存入分布式缓存集群中的某台机器上，当用户访问不同节点时先从缓存中拿Session信息
- 使用场景：集群中机器数多、网络环境复杂
- 优点：可靠性好
- 缺点：实现复杂、稳定性依赖于缓存的稳定性、Session信息放入缓存时要有合理的策略写入

目前生产中使用到的

- 基于tomcat配置实现的MemCache缓存管理session实现(麻烦)
- 基于OsCache和shiro组播的方式实现(网络影响)
- 基于spring-session+redis实现的(最适合)

负载均衡策略

负载均衡策略的优劣及其实现的难易程度有两个关键因素：负载均衡算法，对网络系统状况的检测方式和能力。

1、rr 轮询调度算法。顾名思义，轮询分发请求。

优点：实现简单

缺点：不考虑每台服务器的处理能力

2、wrr 加权调度算法。我们给每个服务器设置权值weight，负载均衡调度器根据权值调度服务器，服务器被调用的次数跟权值成正比。

优点：考虑了服务器处理能力的不同

(http://w

3、sh 原地地址散列：提取用户IP，根据散列函数得出一个key，再根据静态映射表，查处对应的value，即目标服务器IP。过目标机器超负荷，则返回空。

4、dh 目标地址散列：同上，只是现在提取的是目标地址的IP来做哈希。

优点：以上两种算法的都能实现同一个用户访问同一个服务器。

5、lc 最少连接。优先把请求转发给连接数少的服务器。

优点：使得集群中各个服务器的负载更加均匀。

6、wlc 加权最少连接。在lc的基础上，为每台服务器加上权值。算法为： $(\text{活动连接数} * 256 + \text{非活动连接数}) \div \text{权重}$ ，计算出来的值小的服务器优先被选择。

优点：可以根据服务器的能力分配请求。

7、sed 最短期望延迟。其实sed跟wlc类似，区别是不考虑非活动连接数。算法为： $(\text{活动连接数} + 1) * 256 \div \text{权重}$ ，同样计算出来的值小的服务器优先被选择。

8、nq 永不排队。改进的sed算法。我们想一下什么情况下才能“永不排队”，那就是服务器的连接数为0的时候，那么假如有服务器连接数为0，均衡器直接把请求转发给它，无需经过sed的计算。

9、LBLC 基于局部性的最少连接。均衡器根据请求的目的IP地址，找出该IP地址最近被使用的服务器，把请求转发之，若该服务器超载，最采用最少连接数算法。

10、LBLCR 带复制的基于局部性的最少连接。均衡器根据请求的目的IP地址，找出该IP地址最近使用的“服务器组”，注意，并不是具体某个服务器，然后采用最少连接数从该组中挑出具体的某台服务器出来，把请求转发之。若该服务器超载，那么根据最少连接数算法，在集群的非本服务器组的服务器中，找出一台服务器出来，加入本服务器组，然后把请求转发之。

读写分离

MySQL主从配置，读写分离并引入中间件，开源的MyCat，阿里的DRDS都是不错的选择。

如果是对高可用要求比较高，但是又没有相应的技术保障，建议使用阿里云的RDS或者Redis相关数据库，省事省力又省钱。

全文检索

如果有搜索业务需求，引入solr或者elasticsearch也是一个不错的选择，不要什么都塞进关系型数据库。

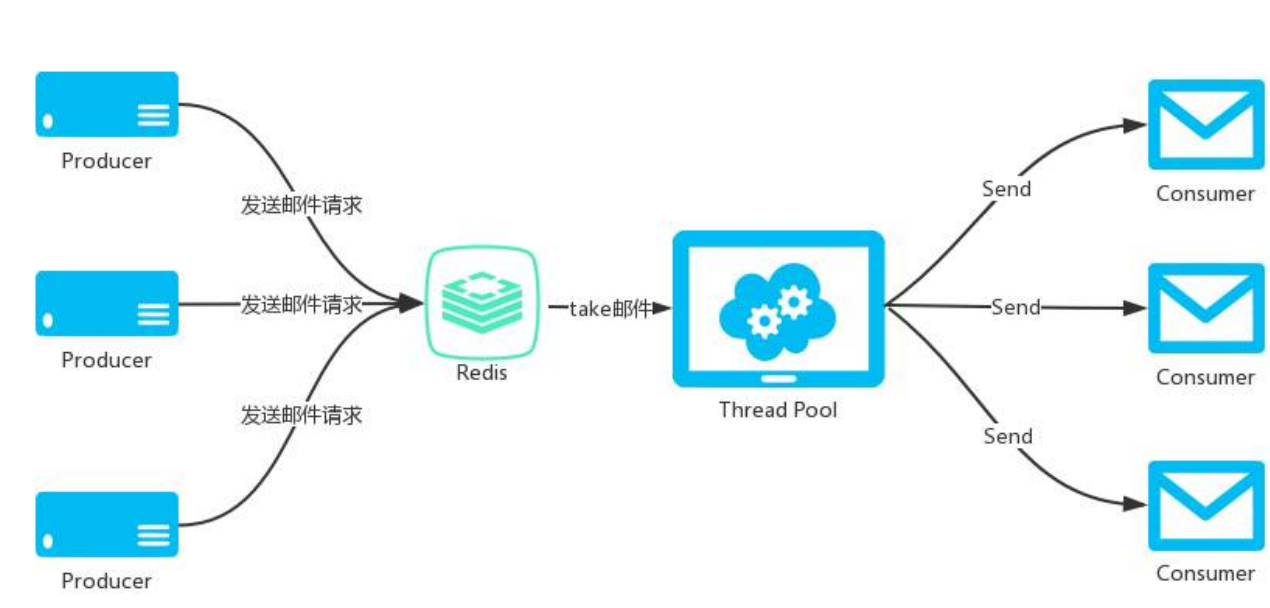
缓存优化

引入缓存无非是为了减轻后端数据库服务的压力，防止其"罢工"。

常见的缓存服务有，Ehcache、OsCache、MemCache、Redis，当然这些都是主流经得起考验的缓存技术实现，特别是Redis已大规模运用于分布式集群服务中，并证明了自己优越的性能。

消息队列

异步通知：比如短信验证，邮件验证这些非实时反馈性的逻辑操作。



流量削锋：应该是消息队列中的常用场景，一般在秒杀或团抢活动中使用广泛。

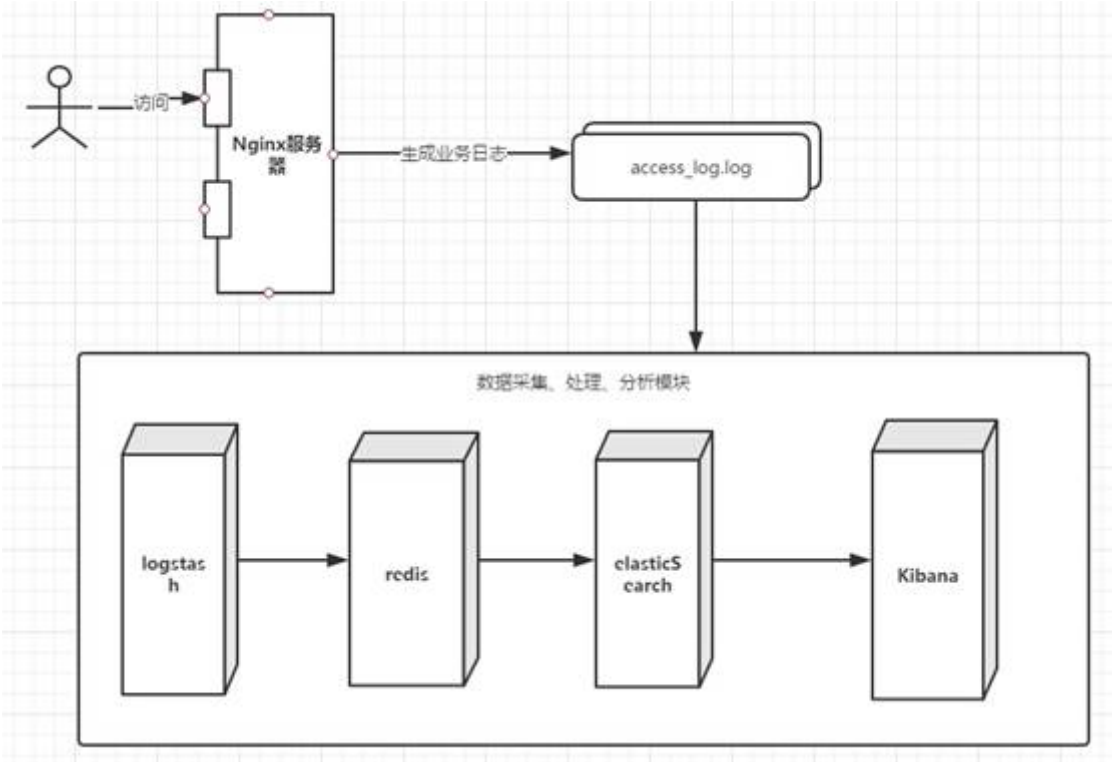
日志处理：系统中日志是必不可少的，但是如何去处理高并发下的日志确是一个技术活，一不小心可能会压垮整个服务。工作中我们常用到的开源日志ELK，为嘛中间会加一个Kafka或者redis就是这么一个道理(一群人涌入和排队进的区别)。

消息通讯：点对点通信(个人对个人)或发布订阅模式(聊天室)。

日志服务

消息队列中提到的ELK开源日志组间对于中小型创业供公司是一个不错的选择。

(http://w



安全优化

以上种种，没有安全做保证可能都会归于零。

- 阿里云的VPN虚拟专有网络以及安全组配置
- 自建机房的话，要自行配置防火墙安全策略
- 相关服务访问，比如MySQL、Redis、Solr等如果没有特殊需求尽量使用内网访问并设置鉴权
- 尽量使用代理服务器，不要对外开放过多的端口
- https配合HTTP/2.0也是个不错的选择

发表评论

邮箱

发表评论

验证码：

发表评论

(http://w

dbaplus社群

线上分享 (http://dbaplus.cn/activity-12-1.html)	专家专栏 (http://dbaplus.cn/blog-56-1.html)
线下沙龙 (http://dbaplus.cn/activity-48-1.html)	免费工具 (http://dbaplus.cn/list-68-1.html)
Gdevops峰会 (http://gdevops.com/)	关于我们 (http://dbaplus.cn/about-61.html)
DAMS峰会 (http://www.dams.org.cn/)	

数据连接未来

dbaplus：数据连接未来！围绕Data、Blockchain、AiOps的企业级专业社群。行业大咖、技术干货，每天精品原创文章推送，每周线上技术分享，每月线下技术沙龙，受众20W+。

原创投稿 | 推广合作 | 媒体合作 | 资源对接 联系人：林禹廷 电话：15876566088 QQ：1134224462
邮箱：linyuting@dbaplus.cn



扫码关注加入各城市微群

