任何一个大型网站都是经历用户积累然后成长，从一台服务器到多台服务器才能架构支撑网站现有的数据、用户、页面请求等。大型网站的系统架构并不是开始设计就具备完整的高性能、高可用、安全等特性。APP开发演变的过程是开发模式，技术架构，设计思想逐渐改变的过程。成熟的系统架构是随着业务扩展而逐步完善出来的，不同的业务特征，其内部结构各有侧重点。例如淘宝，京东，腾讯等他们都有各自的业务特性，尽管我们可以找出其中公用的技术，但最后还是按需索取完成自己的系统。**下面是常见的大型网站系统的演化过程，来认识这些技术和手段。**

1. 最初的架构，应用程序，数据库，文件都部署在一台服务器上

服务器

应用程序

文件

2.应用，数据，文件分离随着业务的扩展，一台服务器不能满足性能需求，故将应用程序，数据库，文件各自部署在独立的服务器上，并且根据服务器的用途卑职不同的硬件，达到最佳的性能效果

文件服务器

应用服务器

应用程序

数据库服务器

文件

3．在硬件优化性能的同时，利用缓存改善网站性能。同时也通过软件经性能优化，大部分的网站系统中，都会利用缓存技术改善系统的性能。缓存实现常见的方式是本地缓存、分布式缓存。淡然还有CDN、反响代理等。

应用服务器

应用程序

数据库服务器

文件服务器

文件

缓存服务器1

4.使用集群改善应用服务器性能。应用服务器作为网站的入口，会承担大量的请求，往往可以通过应用服务器集群来分担请求数。应用服务器前面部署负载均会服务器调度用户请求，根据分发策略将请求分发到多个应用服务器节点。常用的负载均衡技术，硬件有F5，软件的又LVS、Nginx、HAProxy。

5.数据库读写分离和分库分表。随着用户量的增加，数据库成为最大的瓶颈，改善数据库性能常用的手段是进行读写分离以及分表

6.使用CDN和反向代理提高网站性能。

7.应用服务使用分布式文件系统。用户一天天增加，业务量越来越大，产生的文件越来越多，单台的文件服务器已经不能满足需求。需要分布式文件系统支撑。常用的分布式文件系统有NFS

8.使用NoSql和搜索引擎，对于海量数据的查询，我们使用NoSql数据库加上搜索引擎可以达到更好的性能。并不是所有的数据都需要分在关系型数据库中。常用的非关系型数据库有mongodb和redis，搜索引擎有lucene

9.将应用服务器进行业务拆分。随着业务进一步扩展，应用程序变得非常臃肿，这时我们需要将应用程序进行业务拆分。每个业务应用负责相对独立的业务运作。

**总结：**

APP服务端架构通过一系列的拆分和整合，及可以优化公司的整体应用架构，又使APP具有稳定性，低耦合，已拆分等特性，其带来的好处是全方面的：

1.实现WEB端应用和移动端应用分离，使二者彻底解耦，各自独立发展。

2.底层核心的服务基于统一业务规则提供逻辑和数据，核心部分接口在服务器内部不区分PC，无线或其他渠道，避免重复开发，减少核心业务逻辑后期版本迭代过程中的失误率。

3.根据无线本身的特点，支持系统层面的集中处理和业务层面的分散处理。通用逻辑支持插件化扩展，可以根据需要逐步补充；Adapter实现内外部接口的无缝转换，可以针对无线场景，做逻辑增强，数据聚合。

数据库

集群1

集群2

集群..

API网关

APP

展示层

数据层

持久化

数据运算

控制层

协调任务

通用层

业务逻辑

网络策略

安全策略

**微服务1**

通用层

监控

封装

适配层

Adapter1

逻辑

Adapter2

Adapter..

智能服务策略

缓存策略

日志

Adapter3

内部服务

搜索服务

详情

结算

出款

微服务2

微服务…