



INFORMATION SHARE

大型网站技术架构

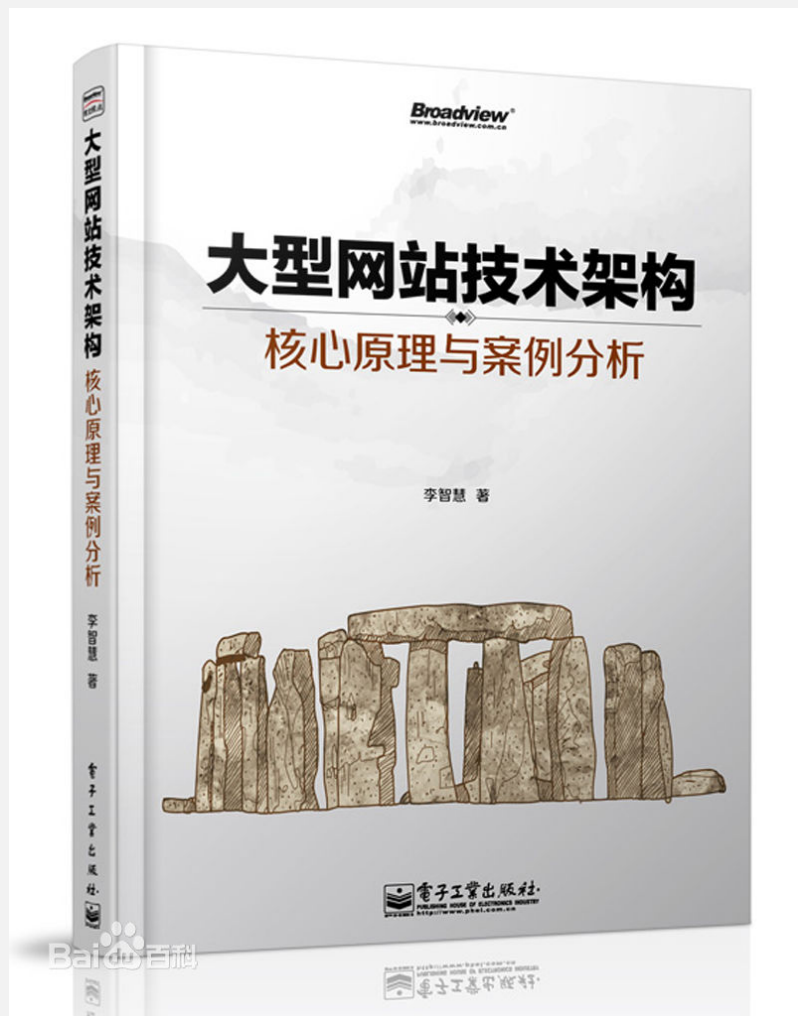
-- 读书分享

白春飞

2018-5-31

BREAD PPT DESIGN

目录 CONTENTS



- 作者是[阿里巴巴](#)网站构建的亲历者李智慧
 - 梳理大型网站技术发展历程
 - 剖析大型网站技术架构模式
 - 讲述大型互联网架构设计的核心原理
 - 典型案例分析

目录 CONTENTS



1

网站的演化

2

大型网站的架构模式

3

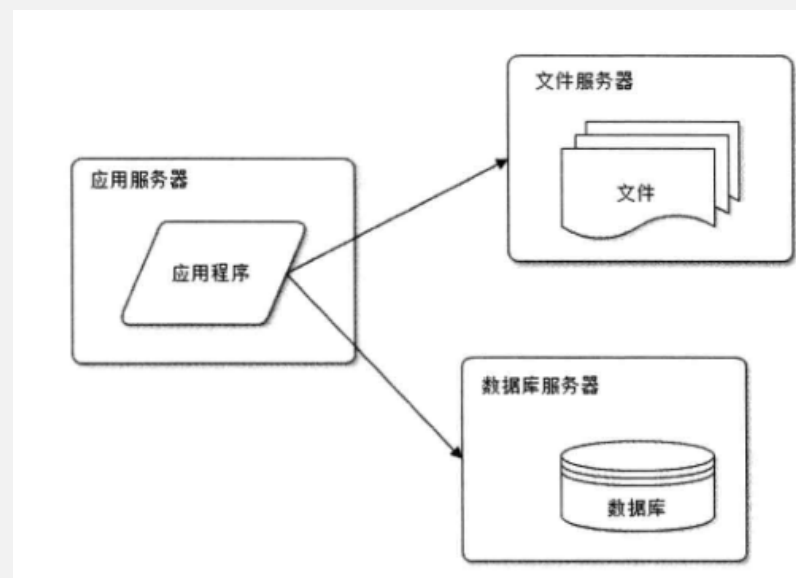
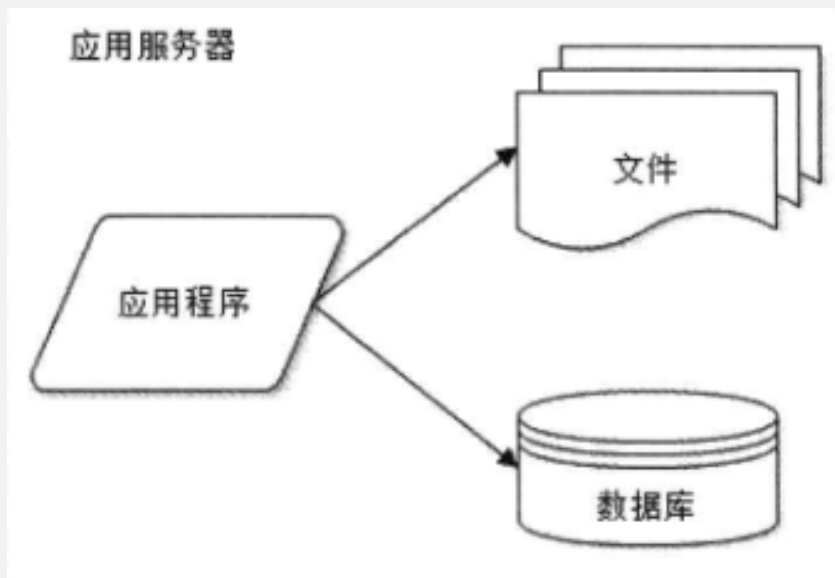
核心要素分析

大型网站软件系统特点？

- 高并发，大流量；
- 高可用：系统7×24小时不间断服务；
- 海量数据：需要存储、管理海量数据，需要使用大量服务器；
- 用户分布广泛，网络情况复杂；
- 安全环境恶劣：由于互联网的开放性，使得互联网更容易受到攻击，大型网站几乎每天都会被黑客攻击；
- 需求快速变更，发布频繁；
- 渐进式发展：几乎所有的大型互联网网站都是从小网站开始，渐进地发展起来的。

大型网站架构变化历程

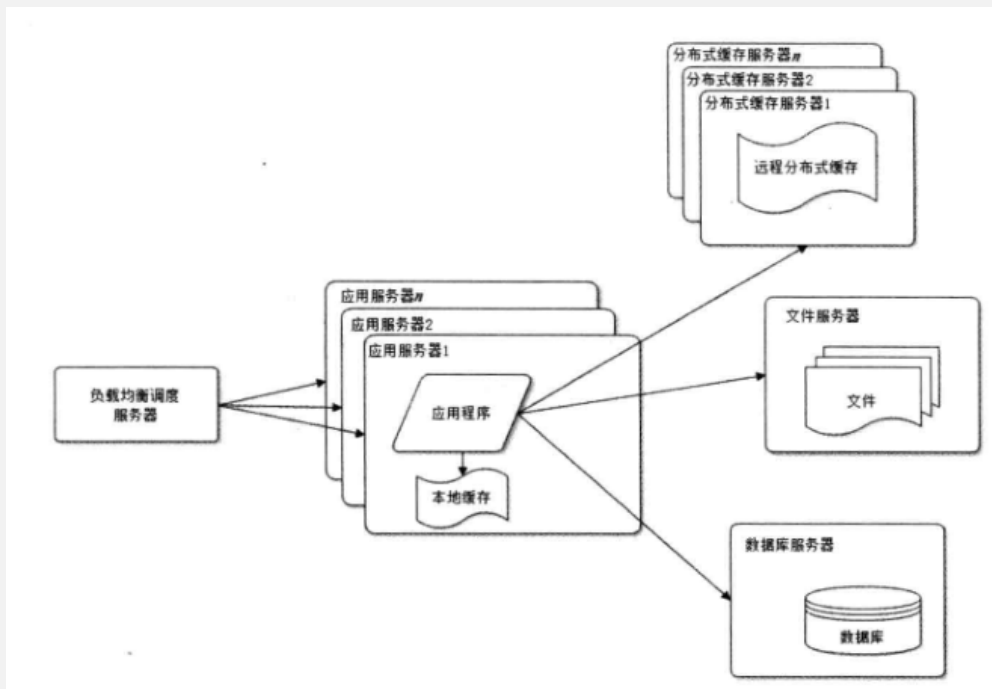
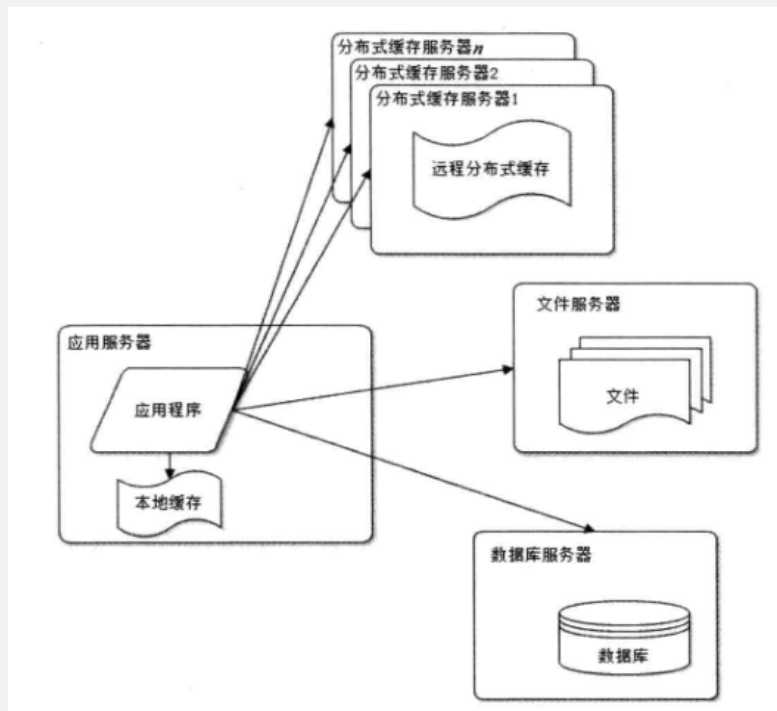
- 初始阶段的网站，一台服务器包含应用、数据库等
- 应用与数据分离



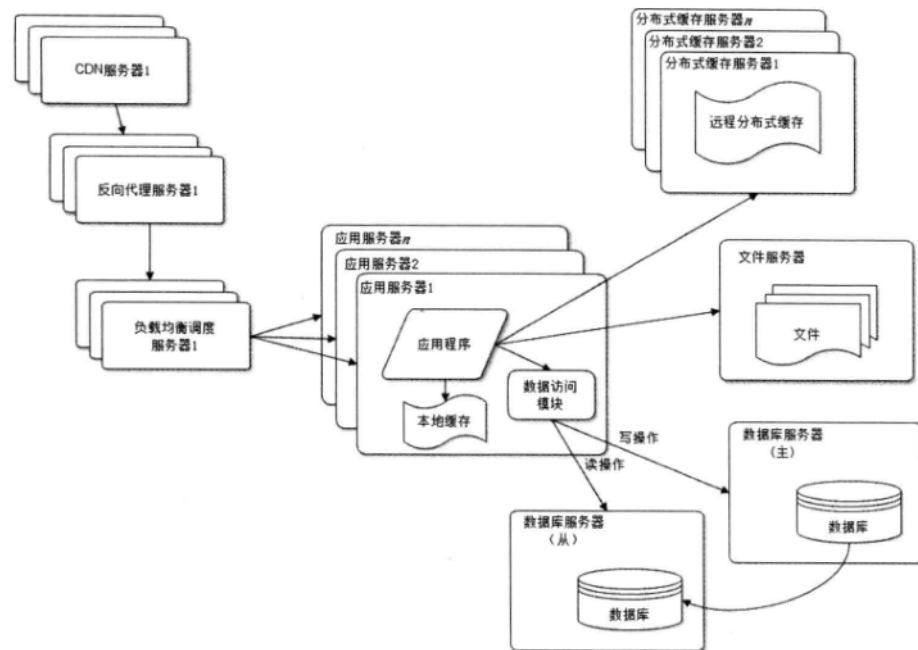
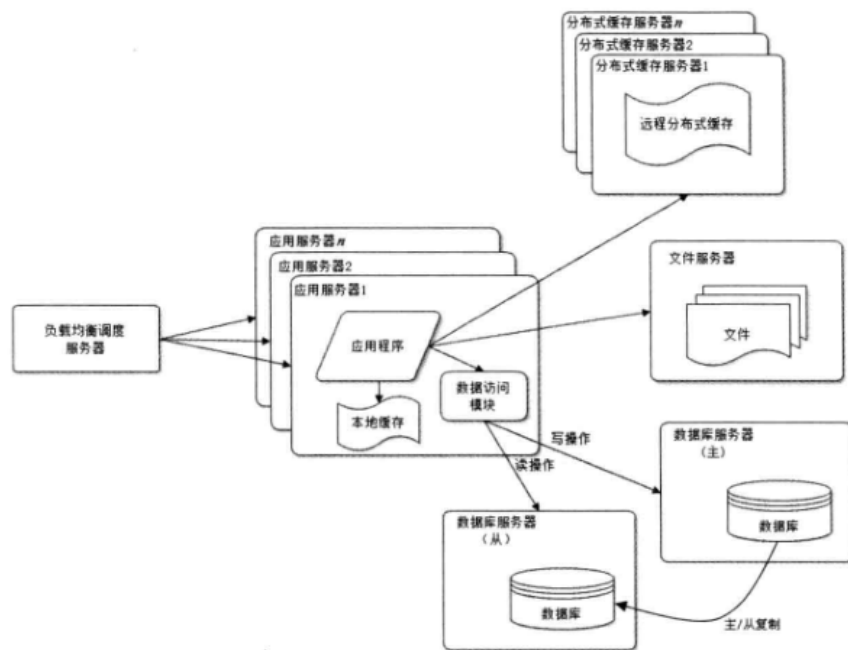
1

网站的演化

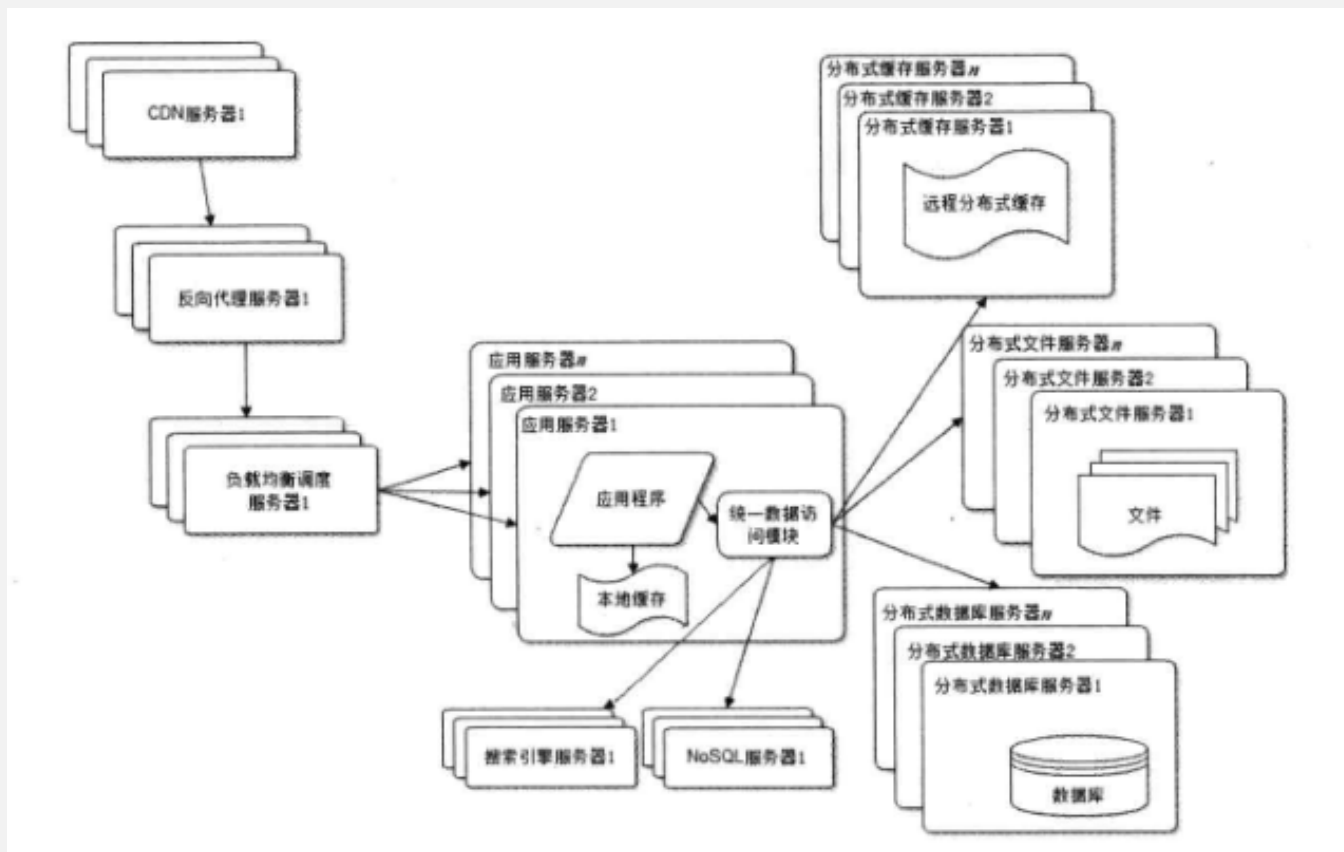
- 使用缓存改善性能
- 使用服务器集群改善网站的并发处理能力



- 数据库读写分离
- 使用反向代理和CDN加速网站相应



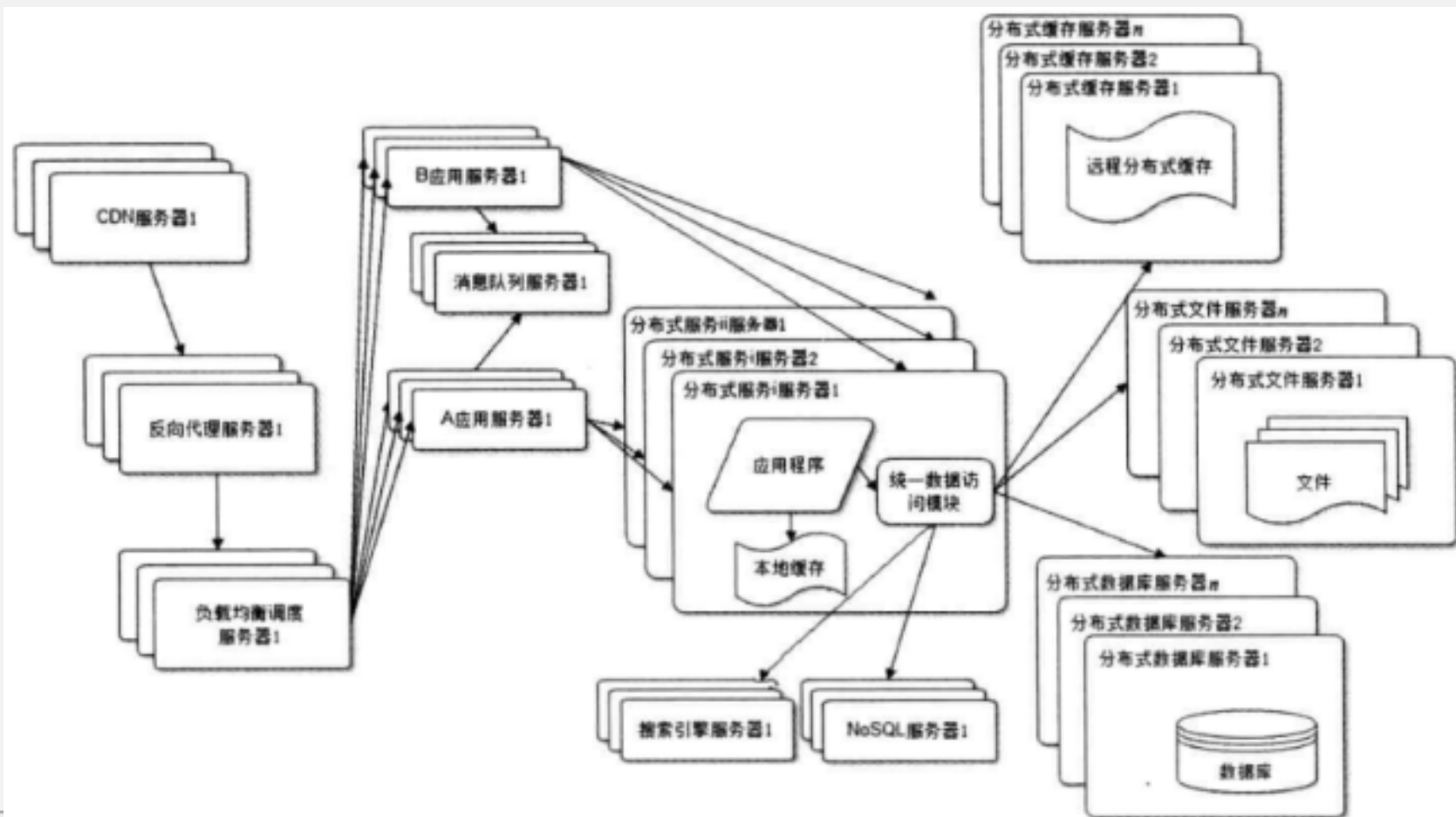
- 使用分布式文件系统和分布式数据库系统
- 使用NoSQL和搜索引擎



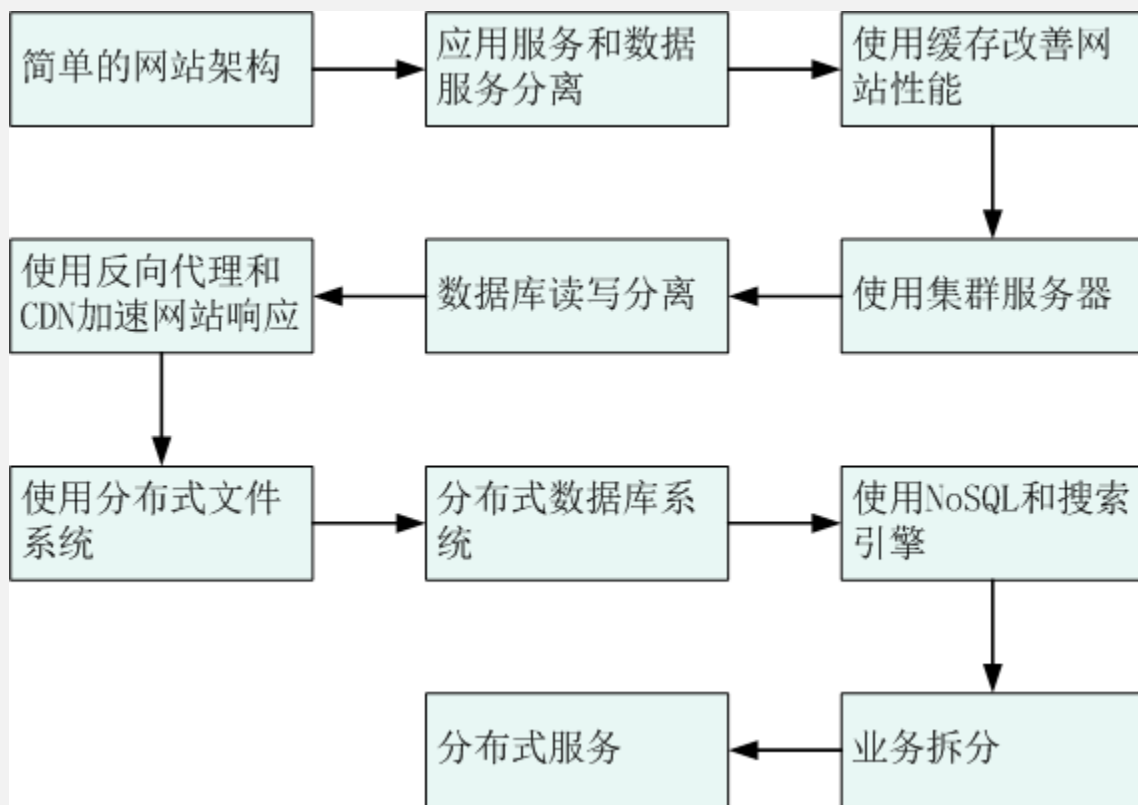
1

网站的演化

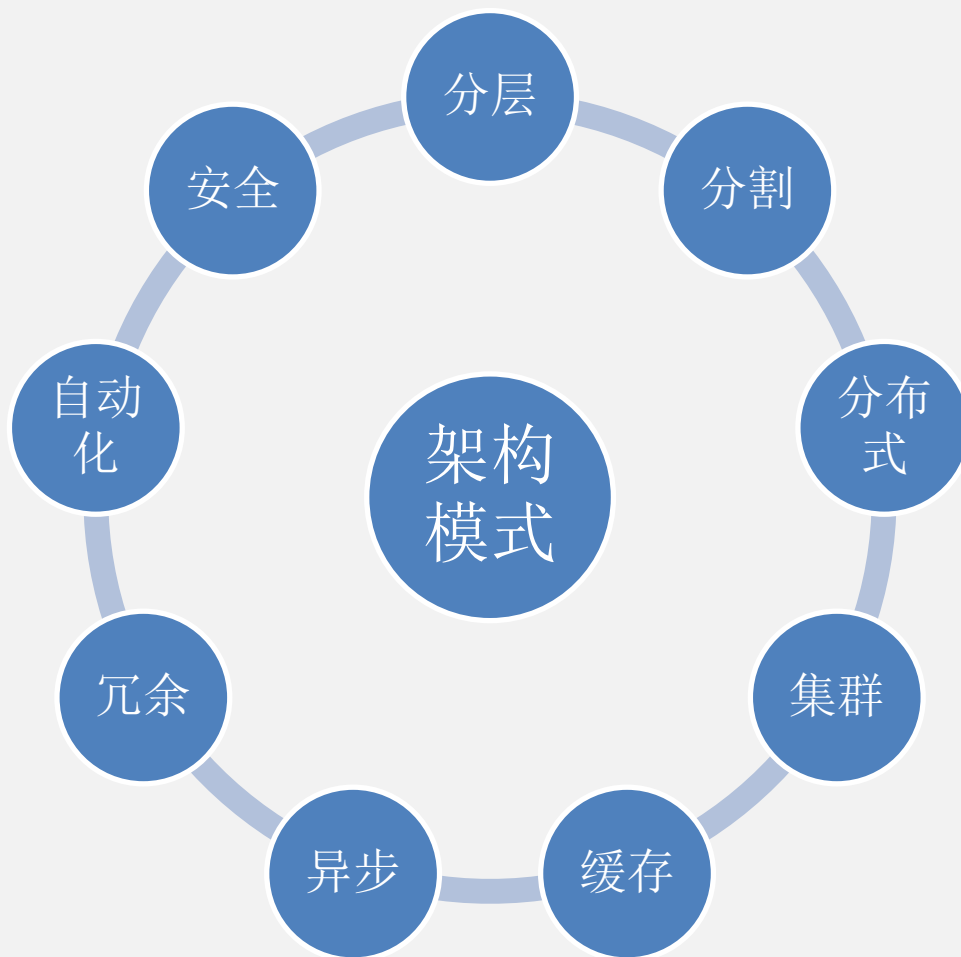
- 业务拆分
- 分布式服务



网站架构的演化



大型网站的架构模式



• 分层

- 将系统横向维度上切分成几个部分，每个部分负责比较单一的职责，通过上层对下层的依赖和调用组成一个完整的系统
- 禁止跨层次的调用和逆向调用

网站分层架构

应用层	负责具体业务和视图展示，如网站首页、搜索输入和结果展示
服务层	为应用层提供服务支持，如用户管理服务，购物车服务等
数据层	提供数据存储访问服务，如数据库、缓存、文件等

- 分割
 - 如果说分层是将软件在横向方面进行切分，那么分隔就是在纵向方面对软件进行切分
 - 大型网站分隔的粒度可能会很小。比如在应用层，将不同业务进行分隔，例如将购物、论坛、搜索、广告分隔成不同的应用，有对立的团队负责，部署在不同的服务器上

● 分布式

- 分布式是指将不同的业务分布在不同的地方
- 分布式应用和服务，把应用和服务模块分布式部署
- 分布式静态资源：把JS，CSS，JPG图片等资源独立分布式部署，并采用独立的域名
- 分布式数据和存储：大型网站的海量数据，单台计算机无法提供空间时，需要分布式部署
- 分布式计算：应用、服务、数据处理都是计算，这些计算量非常庞大，目前很多网站采用hadoop及其MapRedcuce分布式计算框架

● 集群

- 集群是指将多台服务器集中在一起，实现同一业务
- 对于用户访问集中的模块需要将独立部署的服务器集群化，即多台服务器部署相同的应用构成一个集群，通过负载均衡设备共同对外提供服务
- 服务器集群能够为相同的服务提供更多的并发支持，因此当有更多的用户访问时，只需要向集群中加入新的机器即可；另外可以实现当其中的某台服务器发生故障时，可以通过负载均衡的失效转移机制将请求转移至集群中其他的服务器上，因此可以提高系统的可用性

● 缓存

- 缓存目的就是减轻服务器的计算，使数据直接返回给用户。在现在的软件设计中，缓存已经无处不在。具体实现有CDN、反向代理、本地缓存、分布式缓存等
- 使用缓存有两个条件：访问数据热点不均衡，即某些频繁访问的数据需要放在缓存中；数据在某个时间段内有效，不过很快过期，否则会因为数据过期而脏读，影响数据的正确性
- 缓存除了可以加快数据的访问速度，还可以减轻后端应用和数据存储的负载压力

• 异步

- 使用异步，业务之间的消息传递不是同步调用，而是将一个业务操作分成多个阶段，每个阶段之间通过共享数据的方法异步执行进行协作
- 异步调用可以提高系统的可用性，加快网站的响应速度，消除并发访问高峰。但是可能会对用户体验及业务流程造成影响

- 冗余
 - 网站需要 7×24 小时连续运行，那么就得有相应的冗余机制，以防某台机器宕掉时无法访问，而冗余则可以通过部署至少两台服务器构成一个集群实现服务高可用。数据库除了定期备份还需要实现冷热备份。甚至可以在全球范围内部署灾备数据中心

- 自动化
 - 具体有自动化发布过程，自动化代码管理、自动化测试、自动化安全检测、自动化部署、自动化监控、自动化报警、自动化失效转移、自动化失效恢复等

● 安全

1. 通过密码和手机验证码身份认证。
2. 登录、交易等操作需网络通信加密，网站服务器上存储的敏感数据也加密处理。
3. 使用验证码识别，防止机器人程序滥用网络资源攻击网站。
4. 对常见的用于攻击网站的XSS攻击、SQL注入进行编码转换等处理。
5. 对垃圾信息、敏感信息过滤。
6. 对交易转账等重要操作根据交易模式和交易信息进行风险控制。

3

核心要素

一

高性能

二

可用性

三

伸缩性

四

扩展性

五

安全性

- 判断一个网站性能好不好有哪些指标
 - 响应时间，从发出请求到收到数据所需要的时间
 - 并发数，系统能同时处理请求的数目
 - 吞吐量，单位时间内，系统能处理的请求数量
 - 性能计数器，一些数据指标，包括对象与线程数，内存使用，CPU使用等

- 性能测试方法
 - 性能测试
 - 负载测试
 - 压力测试
 - 稳定性测试

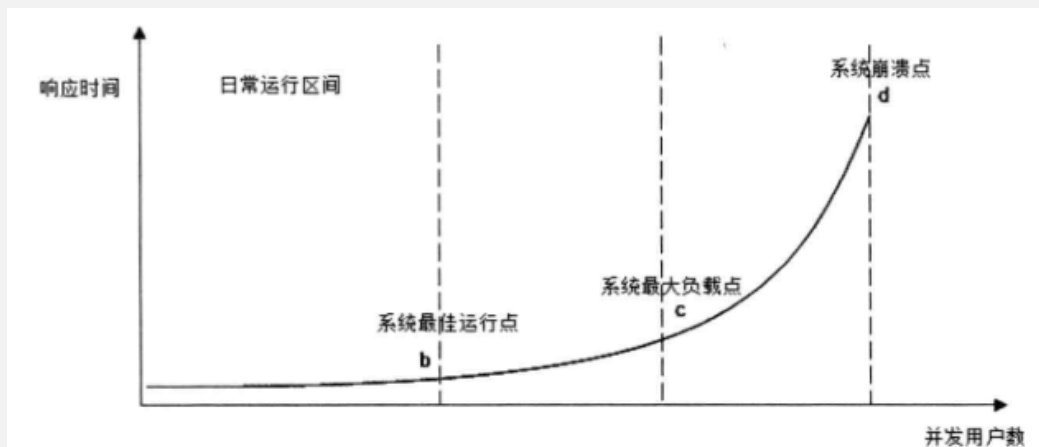
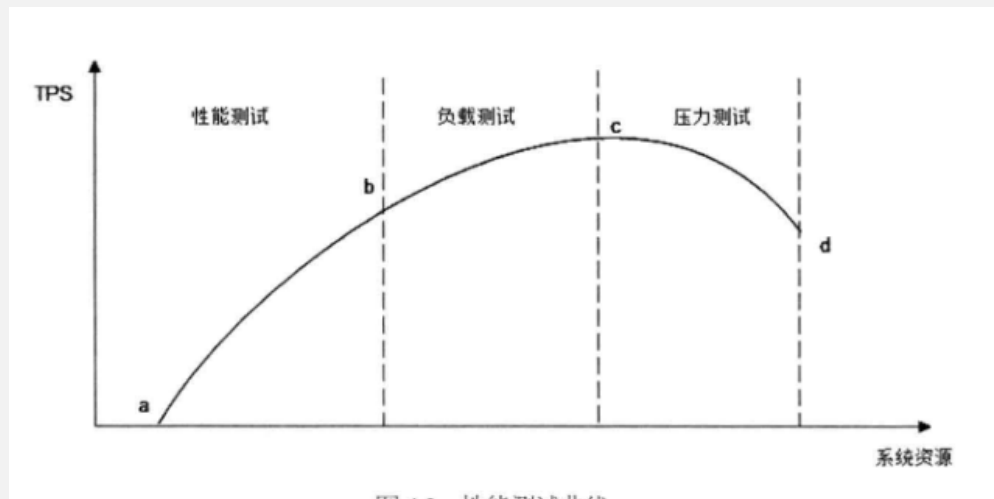




图 4.4 并发用户访问响应时间曲线

1

 WEB前端性能优化

2

 应用服务器性能优化

3

 存储性能优化

- Web前端优化

- 浏览器端优化

- 减少http请求，合并CSS,JS,图片等
 - 使用浏览器缓存，静态资源可以缓存在浏览器中
 - 启用压缩，对文件压缩，减少传输的数据量，浏览器端解压，这对服务器和浏览器端都有压力
 - CSS放在页面最上，js放在页面最下面
 - 减少Cookie传输

- Web前端优化

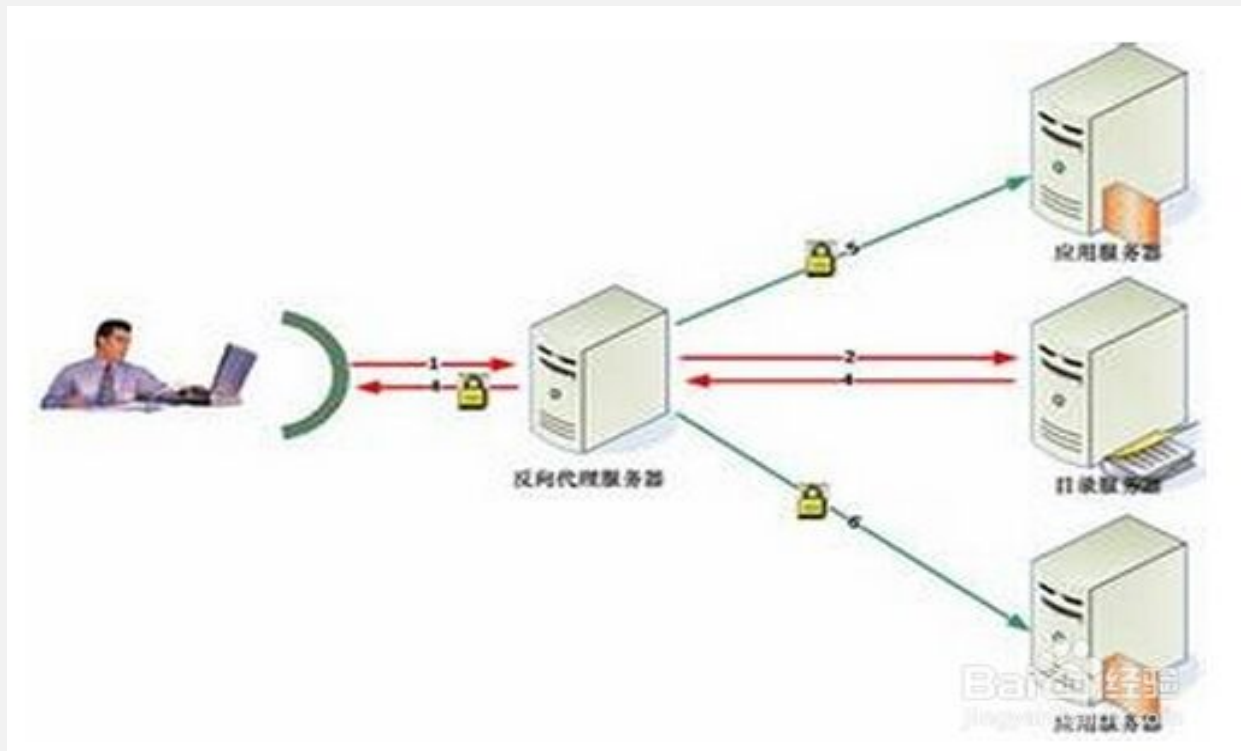
- CDN加速

- CDN本质就是缓存，而且将数据缓存在离用户最近的地方。
CDN就部署在网络营运商的机房中

- 反向代理

- 传统代理服务器位于浏览器一侧，而反向代理服务器位于网站机房一侧。反向代理服务器通过配置缓存功能，加速网站响应
 - 另外，反向代理具有保护网站安全的作用，所有请求都须经过反代服务器。反代还可以实现负载均衡

— 反向代理



- 应用服务器性能优化

- 分布式缓存

- 分布式缓存架构有两种，一种以Jboss 为代表需要更新同步的分布式缓存，一种以Memcached为代表的互相通信的分布式缓存

- 异步操作

- 使用集群

- 存储性能优化（硬件）

- 高性能的代码

- 多线程，需要注意线程安全问题
 - 资源复用
 - 数据结构
 - 垃圾回收

3

高可用性

1



高可用的应用

2



高可用的服务

3



高可用的数据

4



软件质量保证

5



网站运行监控

- 高可用的应用
 - 通过负载均衡进行无状态的失效转移
 - Session管理
 - 早期网站使用session复制，在集群中同步
 - Session绑定在某台服务器上，但是一旦宕机，session就不存在
 - 利用cookie记录session，保存在客户端上，但是受限于cookie的大小和用户是否开启cookie
 - Session服务器，部署独立的session服务器集群，每次读写都访问session服务器

- 高可用的服务

- 分级管理

- 核心应用和服务优先的使用更好的硬件，其他次之

- 超时设置

- 服务调用时间设置超时时间

- 异步调用

- 通过消息队列方式完成

- 服务降级

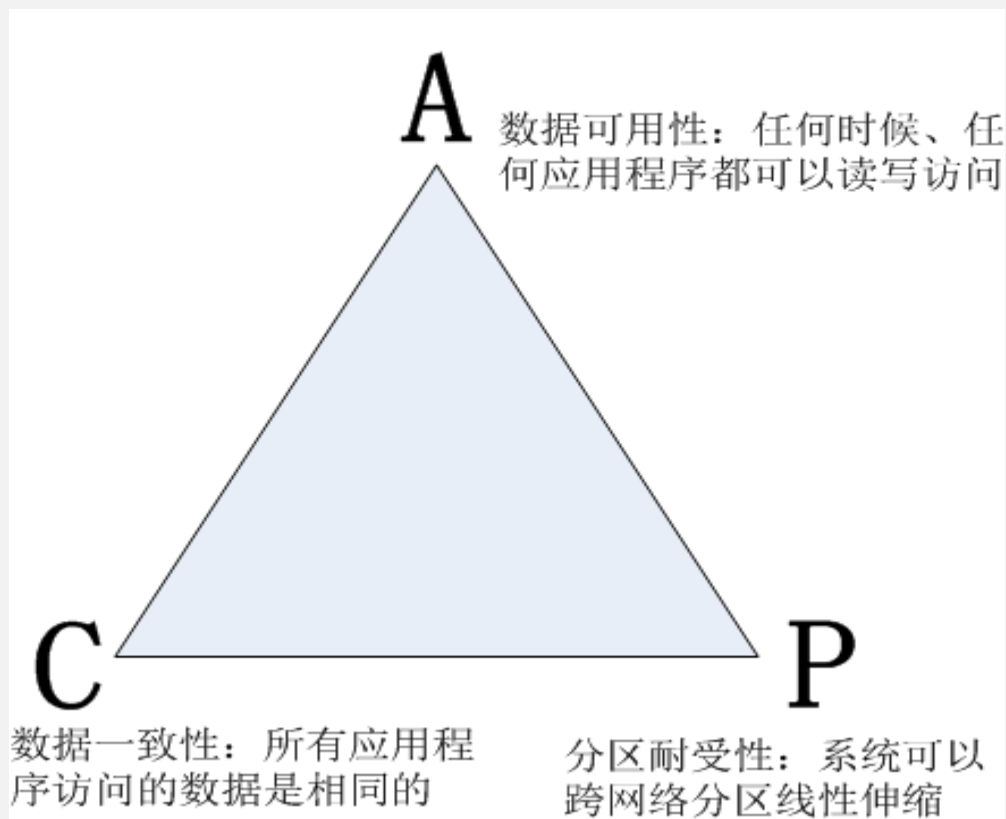
- 网站访问高峰期，可对非重要服务降级：拒绝服务和关闭服务

- 幂等性设计

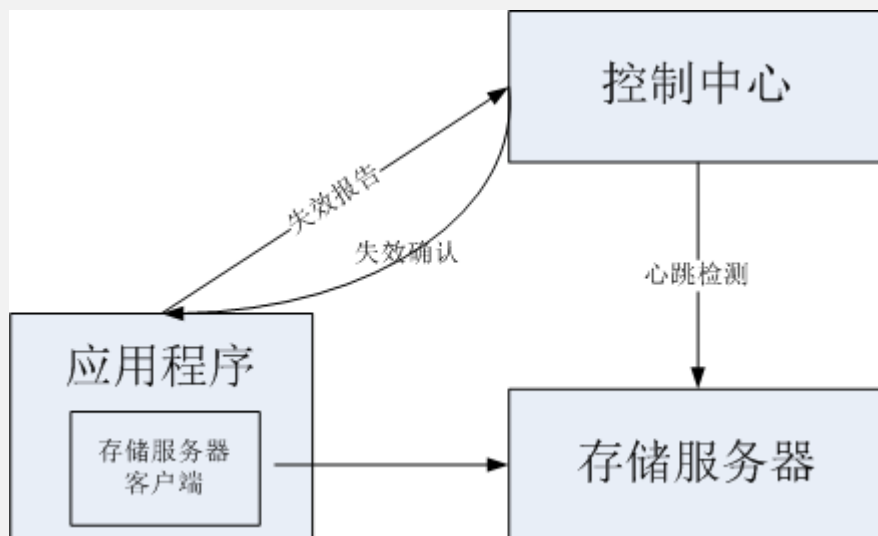
- 高可用的数据

- CAP原理

- 数据持久性 Partition Tolerance
 - 数据可访问性 Availability
 - 数据一致性 Consistency
 - 数据强一致，数据更新结果和操作响应总是一致的
 - 数据用户一致，数据在存储中可能是不一致的，但是用户访问时，通过就错和校验，可以确定一个一直的数据给用户
 - 数据最终一致，物理存储的数据可能是不一致的，用户访问时也可能是
不一致的，但是经过一段时间后，数据最终会达到一致。



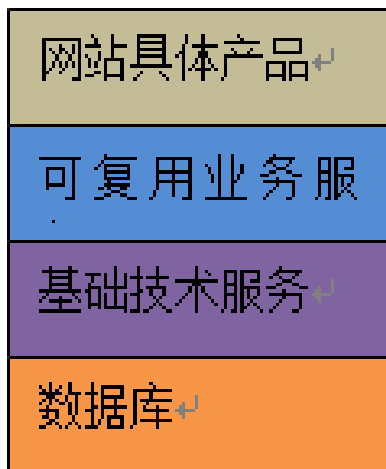
- 高可用的数据
 - 数据备份
 - 失效转移



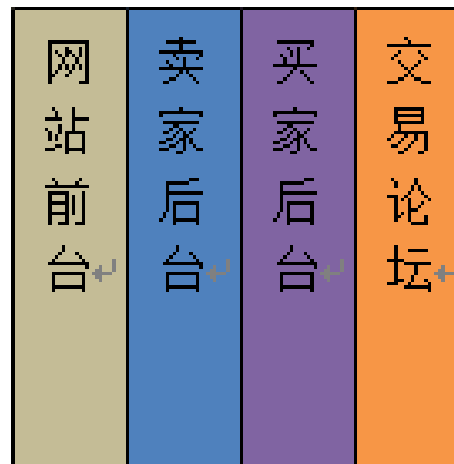
- 运行监控
 - 监控数据采集
 - 用户行为日志收集
 - 服务器性能监控
 - 运行数据报告
 - 监控管理
 - 系统报警
 - 失效转移
 - 自动优雅降级

不同功能物理分离

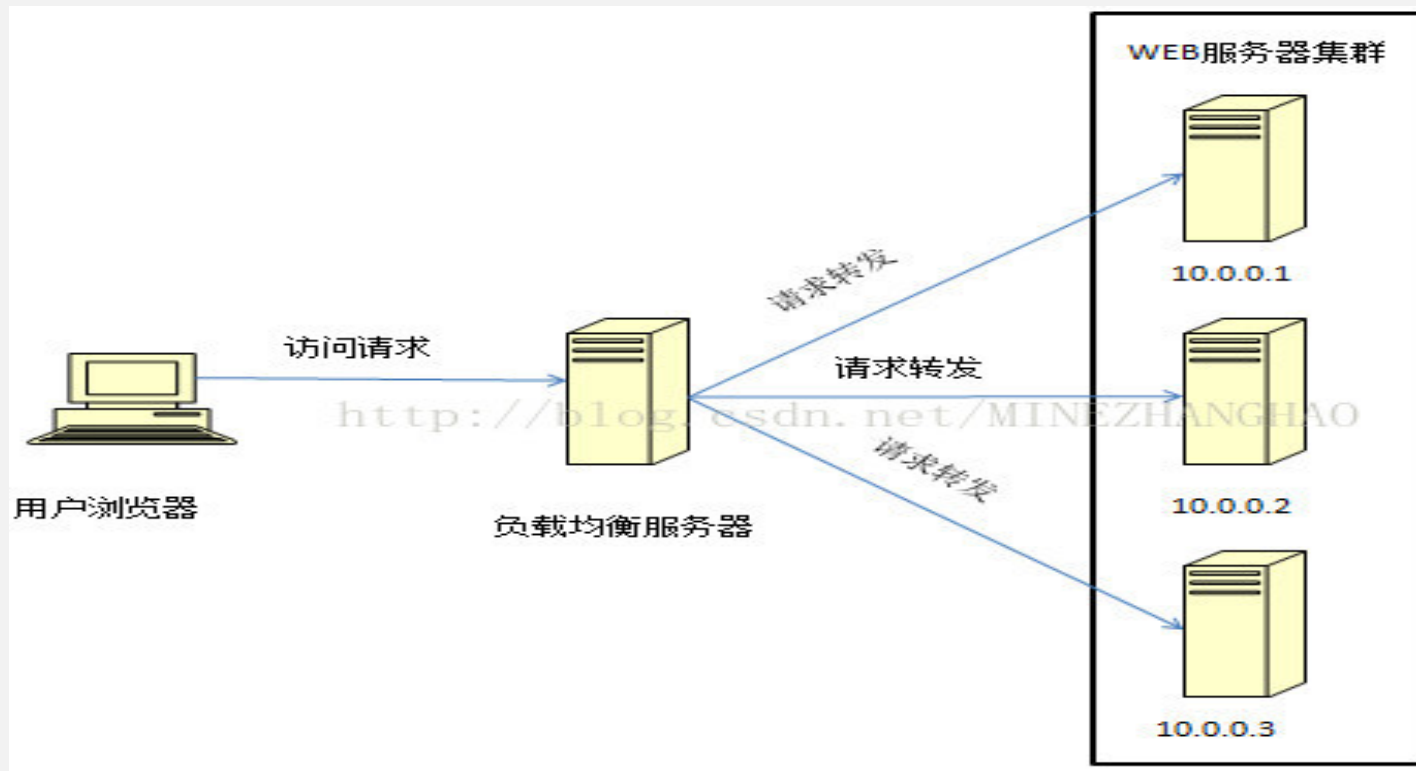
- 纵向分离



- 横向分离



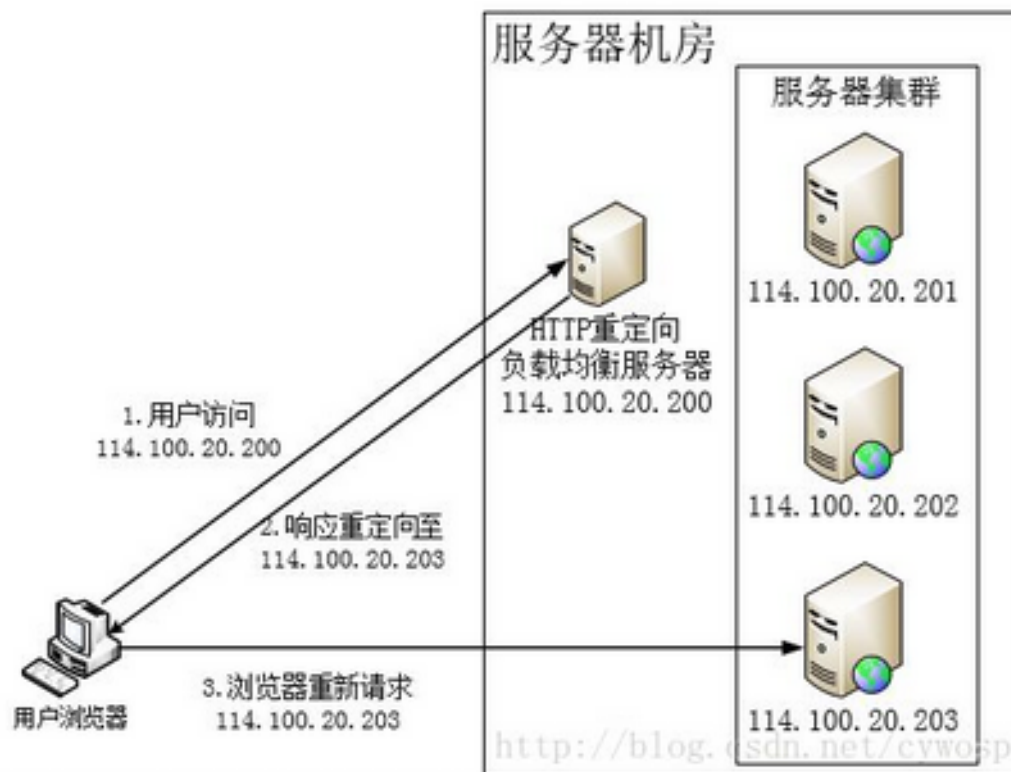
单一功能通过集群实现伸缩



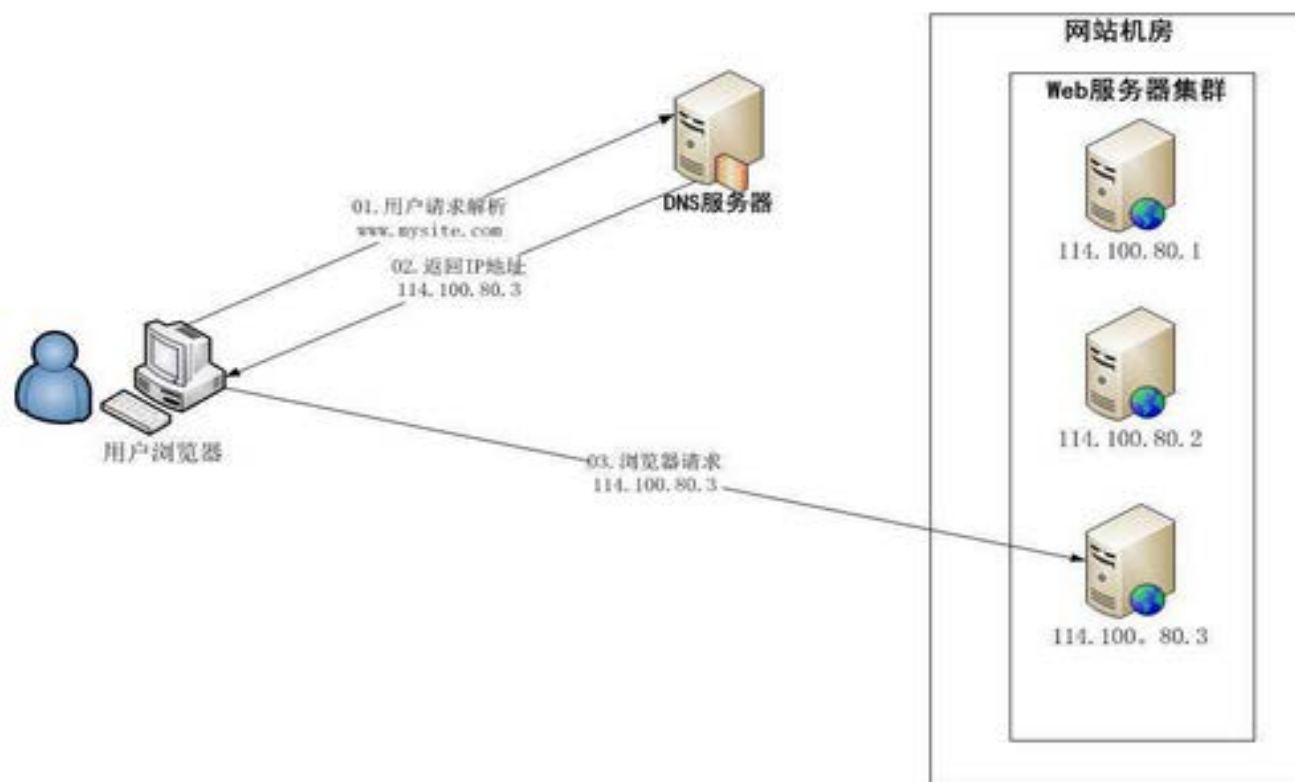
集群的伸缩性设计：

- HTTP重定向负载均衡
- DNS域名解析负载均衡
- 反向代理负载均衡
- IP负载均衡
- 数据链路层负载均衡
- 负载均衡算法简介

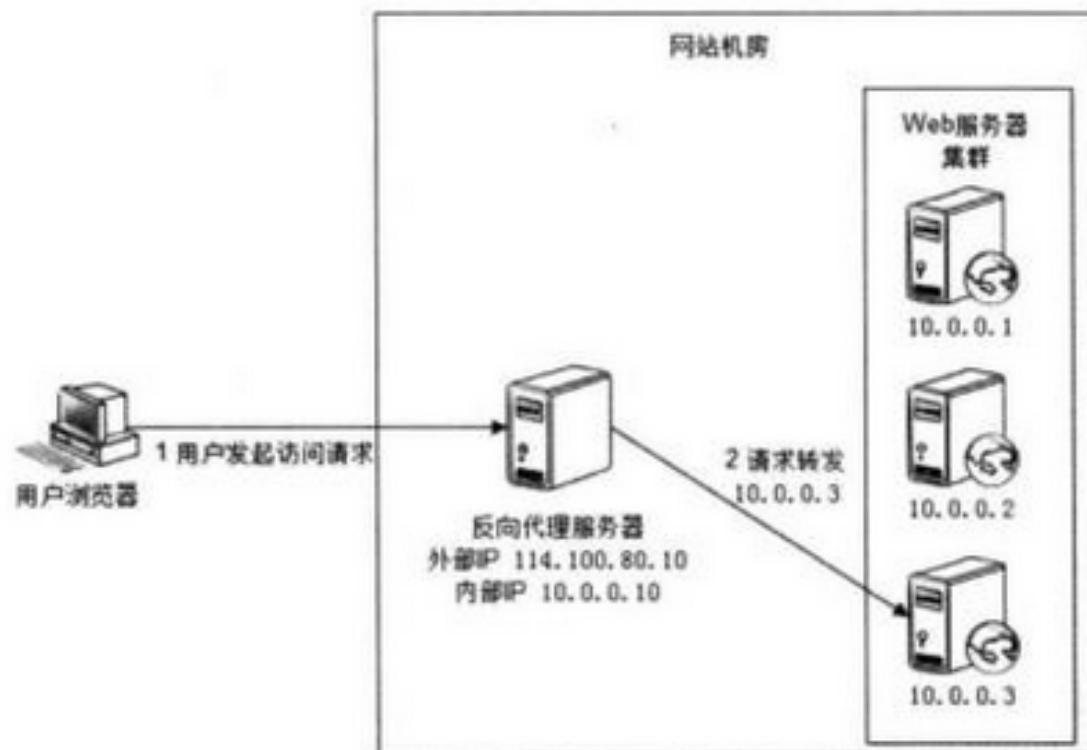
http重定向负载均衡



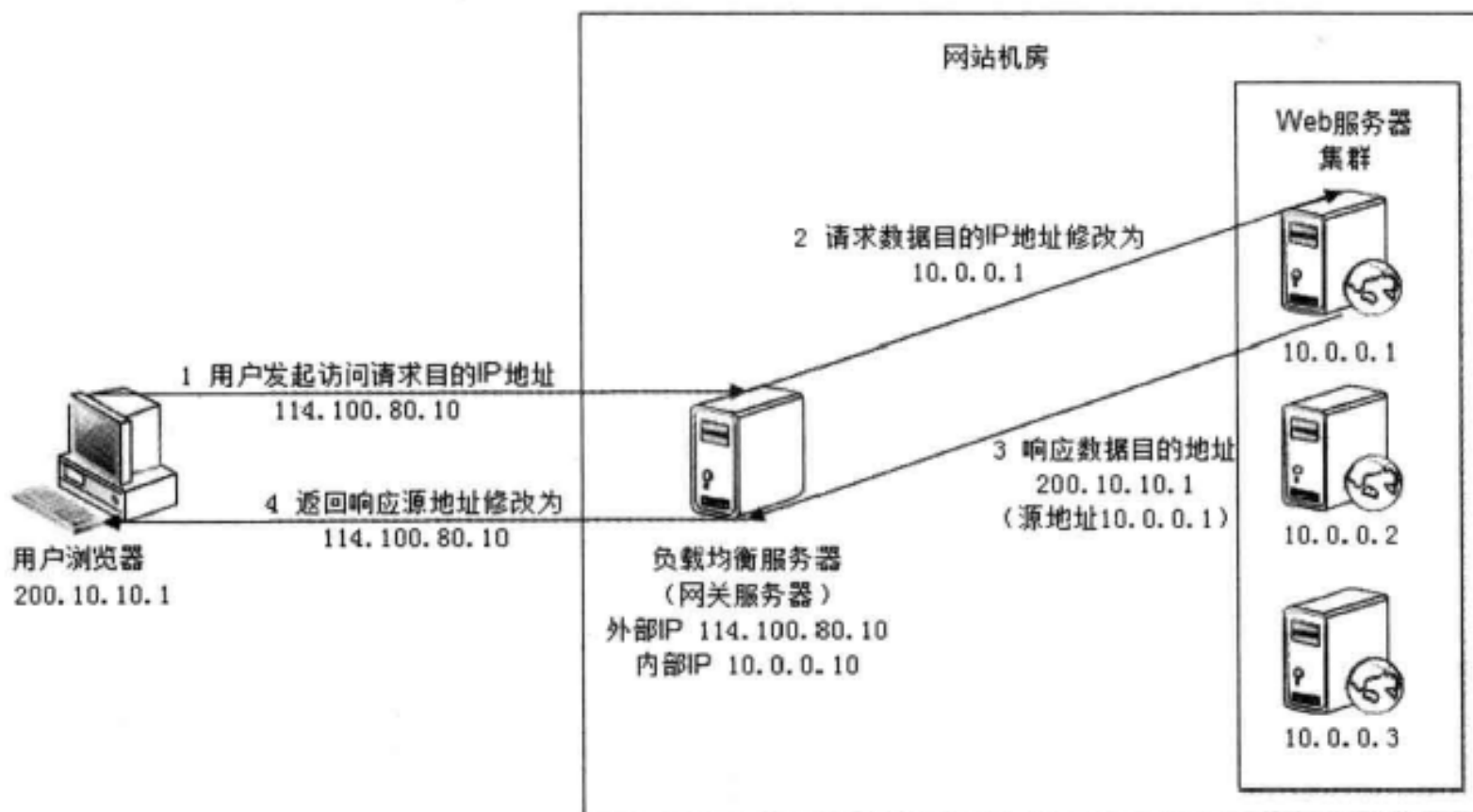
DNS域名解析负载均衡



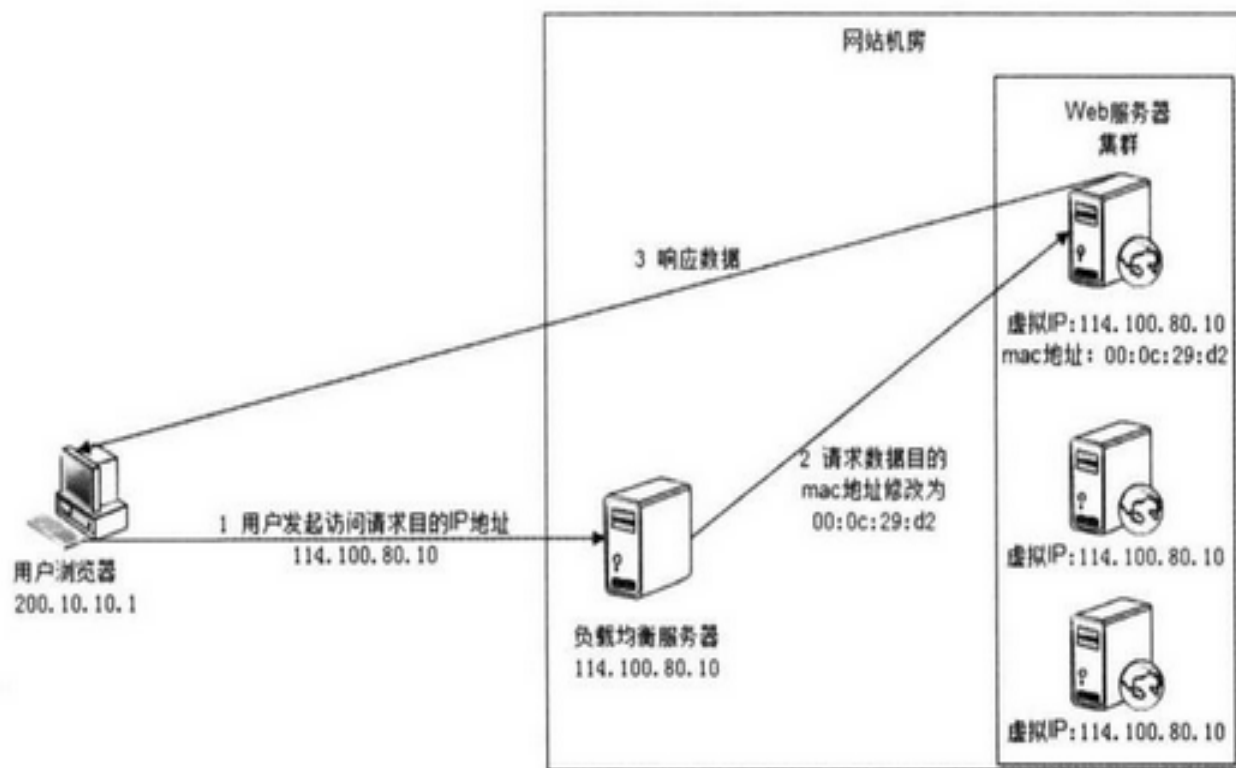
反向代理负载均衡



IP负载均衡



数据链路层负载均衡



数据存储服务器集群的伸缩性设计

- 关系数据库集群的伸缩性设计
- NoSQL数据库的伸缩性设计

3

高可扩展性

1

● 分布式消息队列降低系统耦合性

2

● 分布式服务

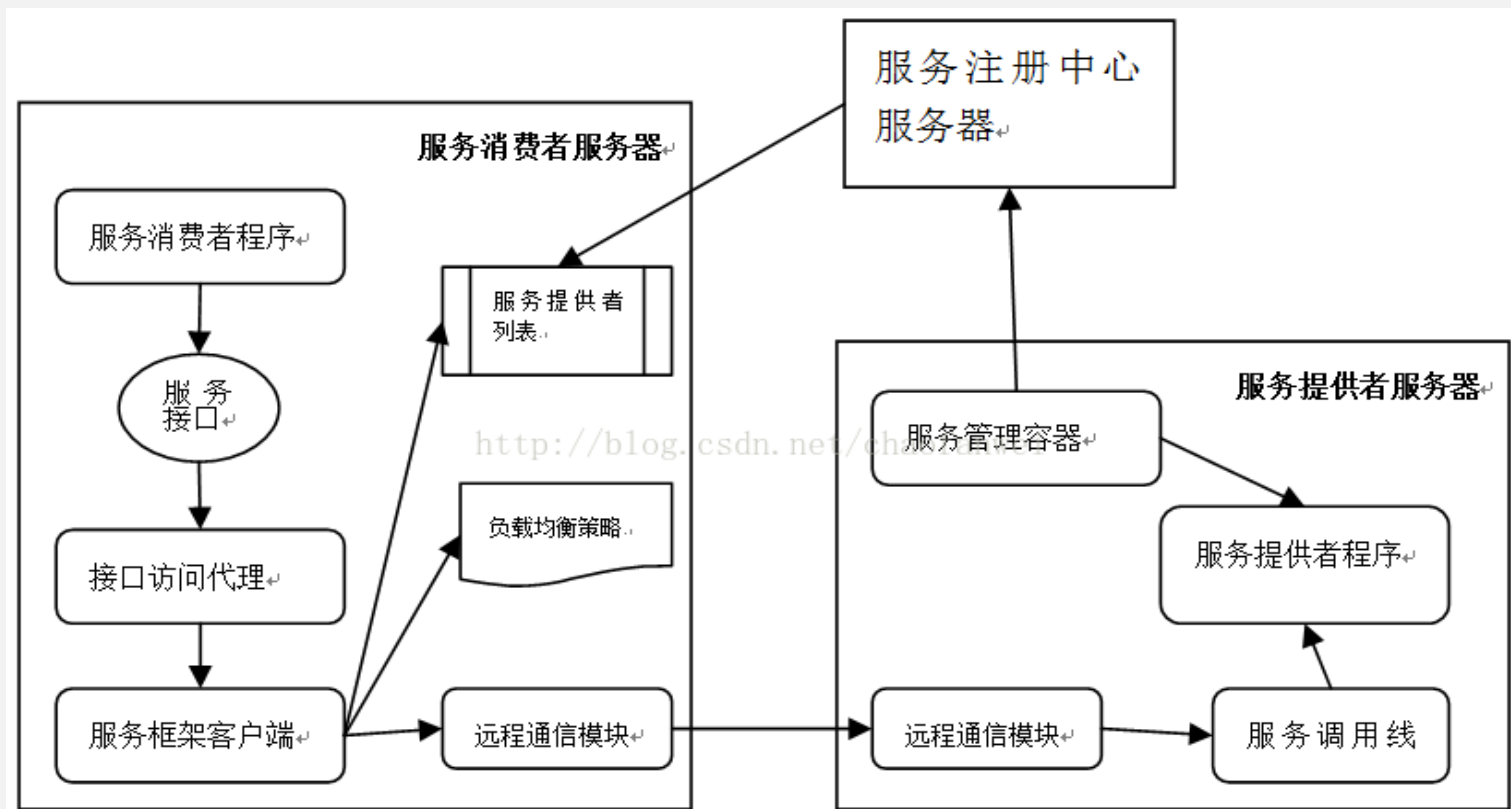
3

● 可扩展的数据结构

利用分布式服务打造可复用的业务平台

- 负载均衡
- 失效转移
- 高效的远程通信
- 整合异构系统
- 对应用最少侵入
- 版本管理
- 实时监控

- 目前国内有较多成功实施案例的开源分布式服务框架是阿里巴巴的Dubbo，



面临的威胁

XSS攻击 跨站点脚本攻击

SQL注入攻击

CSRF攻击 跨站点请求伪造

其他攻击和漏洞

谢谢！