

# “大型” 网站技术架构探讨

余浩东  
2011年6月

# 大型网站架构的目标与挑战

网站架构演变及其技术脉络

架构设计理论与原则

讨论及总结

# 大型网站架构的目标与挑战

## ■何谓“大型”网站？

💡 没有统一的判断标准，流量大小是一个重要指标

网站	日均流量[IP/PV]	
www.hao123.com	IP≈ 5,972,587 PV≈ 9,376,962	
www.facebook.com	IP≈229,680,000 PV≈2,955,981,600	
www.sina.com.cn	IP≈25,680,000 PV≈222,132,000	
www.tianya.cn	IP≈5,532,000 PV≈25,723,800	
<i>www.pingan.com</i>	<i>IP≈300,000 PV≈747,000</i>	

### pingan.com的搜索结果



### javaeye.com的搜索结果

相关搜索：[javaeye](#)



💡 日均流量至少IP>1,000,000才算大型网站

# 大型网站架构的目标与挑战

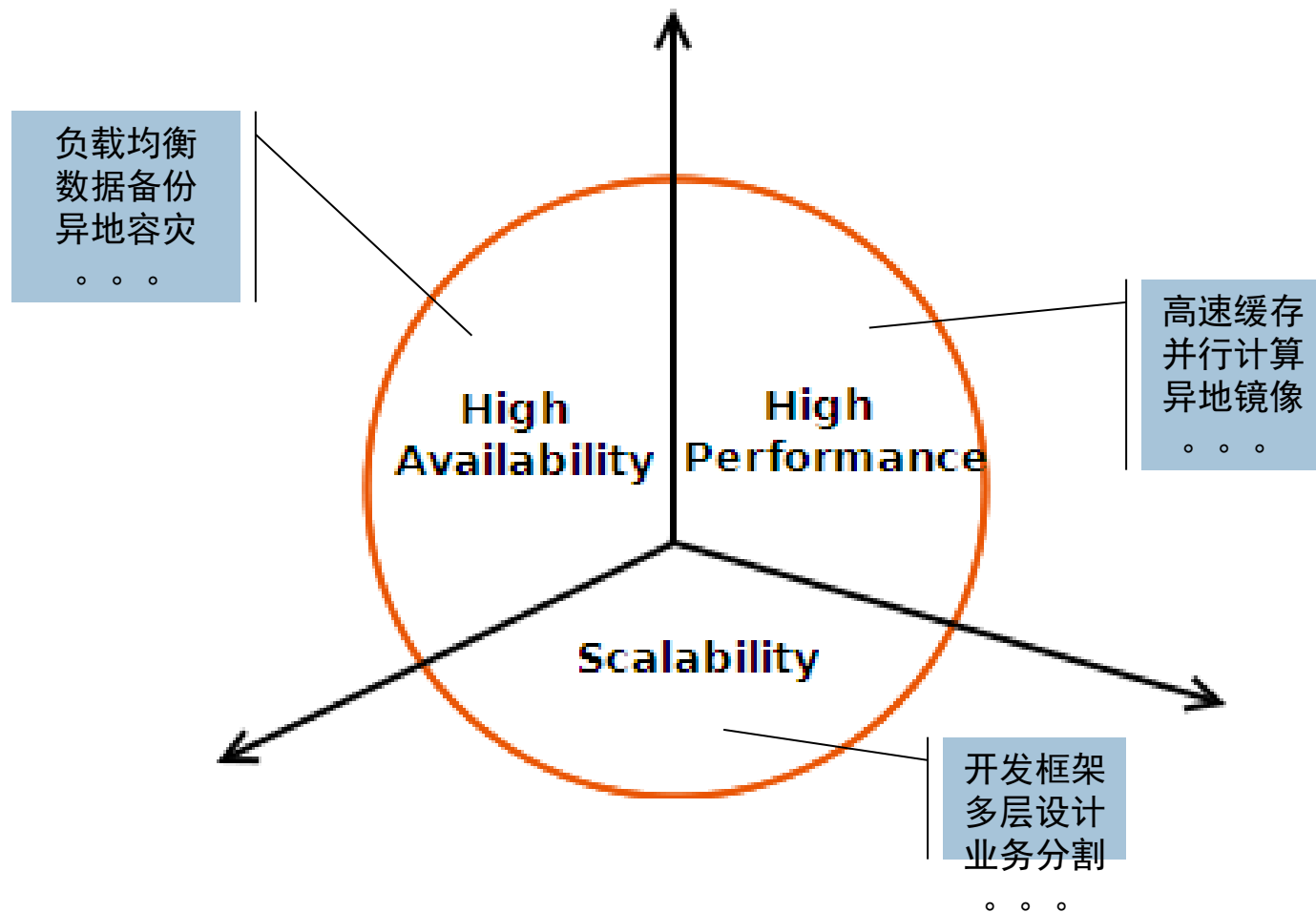
## ■何谓“大型”网站？



网站内容是否“动态”才是关键

# 大型网站架构的目标与挑战

## ■ 网站架构目标与挑战



🤔 每个目标后面面临着技术、设计、维护等诸多方面的挑战。

而目标本身的期望值也会根据实际情况进行调整，这也意味着**网站架构建设是个不断调整的过程。**

**大型网站架构的目标与挑战**

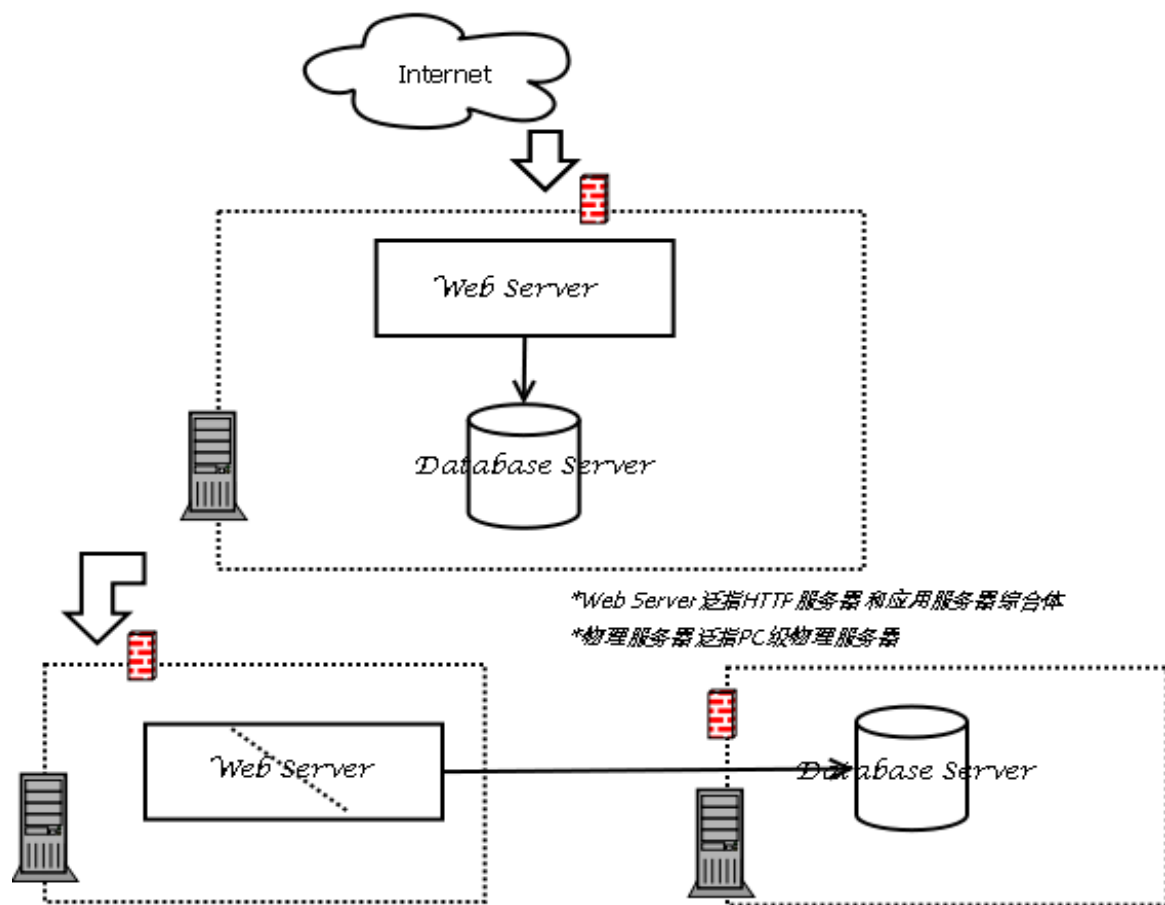
**网站架构演变及其技术脉络**

**架构设计理论与原则**

**讨论及总结**

# 网站架构演变及其技术脉络

## ■[Step1]Web动静态资源分离及其与DB物理分离



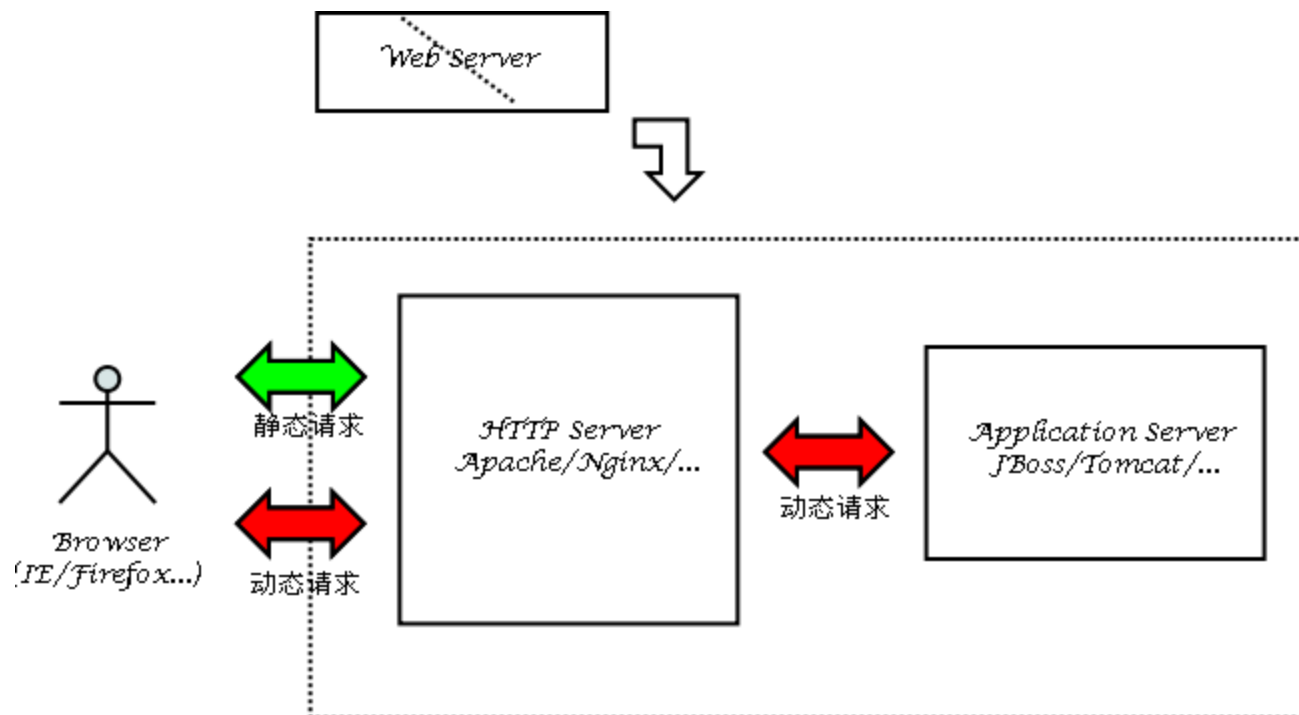
■优点：“简单”、安全性提高

■缺点：存在单点，谈不上高可用性（high availability架构目标）

■技术点：应用设计要保证可扩展（framework很重要Spring/Beetle）、Web Server动/静态资源分离  
Web Server（Apache\Nginx\IIS\JBoss...）、  
Database Server（Mysql\Oracle\Redis...）

# 网站架构演变及其技术脉络

## ■[Step1]技术点—Web动静态资源分离

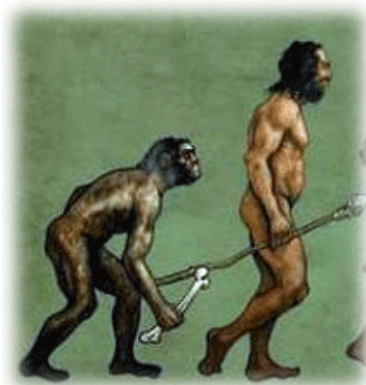
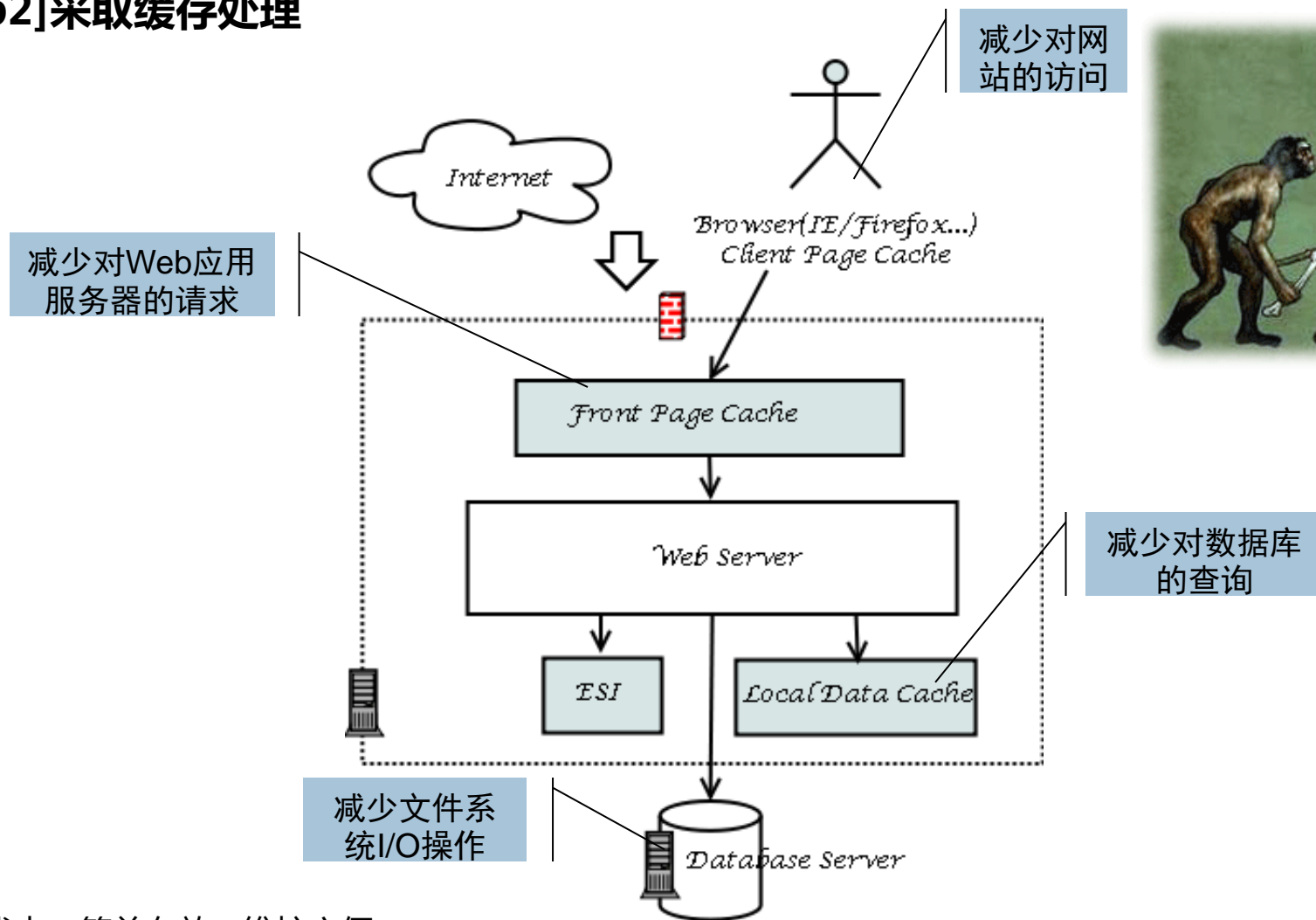


img, doc, js, css等静态资源使用单独的Web HTTP Server处理请求  
动态页面静态化处理



# 网站架构演变及其技术脉络

## ■[Step2]采取缓存处理



■优点：简单有效、维护方便

■缺点：依然存在单点

■技术点：客户端（浏览器）缓存、前端页面缓存、页面片段缓存、本地数据缓存/数据库缓存

# 网站架构演变及其技术脉络

## ■[Step2]技术点—客户端（浏览器）缓存

### 技术点说明

根据HTTP协议特性，修改Header参数（Cache-Control、Expires、Pragma、Last-Modified、Etag），让浏览器来缓存页面（一些优秀开发框架会对此做透明的封装，例如：Beetle）<http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html>

使用HTTP1.1协议，由于http pipelining技术特性，能够使用get请求的决不采取post请求

为了节约带宽，压缩页面（Content-Encoding: gzip）；页面各个元素能“小”即“小”，例如：js包压缩，js合并，图片压缩等

会话状态信息采取Cookie代替传统使用服务器Sessions对象存储习惯做法；使用Ajax实现页面局部刷新

如果可能，可采取浏览器插件技术突破浏览器功能限制，将原本在服务器端运算，尽量迁到浏览器端。ActiveX/Applet/Flash/....  
HTML5 🤖 最值得期待，她的出现必定改变整个Web世界

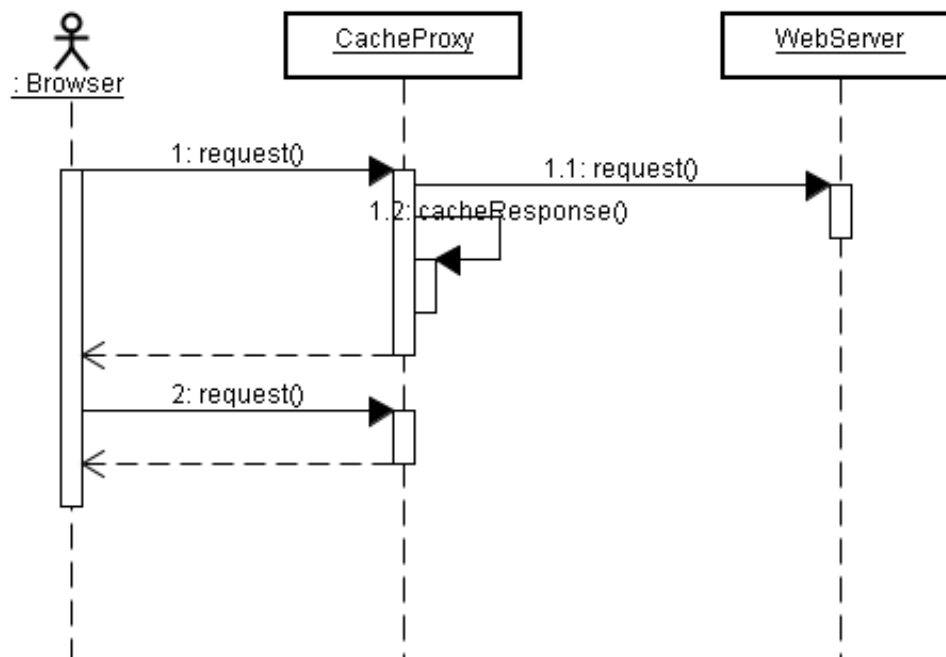
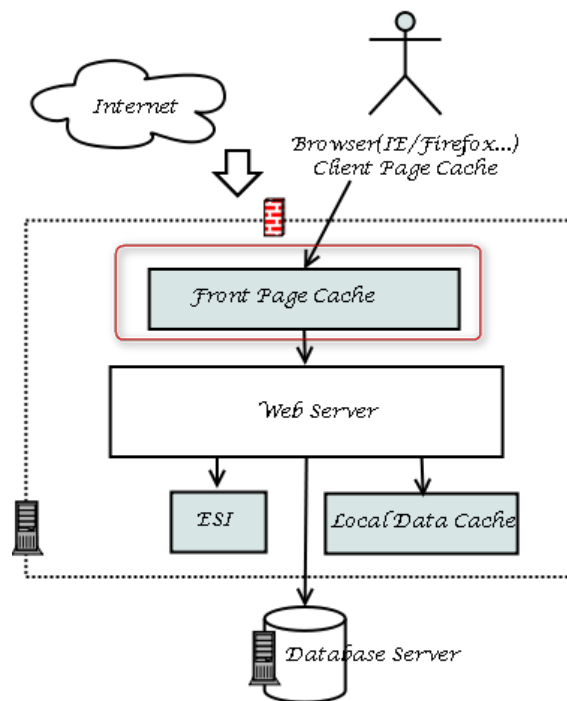


能够让浏览器缓存的数据一定要缓存；浏览器能够处理的运算，决不放在服务器端来处理。



# 网站架构演变及其技术脉络

## ■[Step2]技术点—前端页面缓存

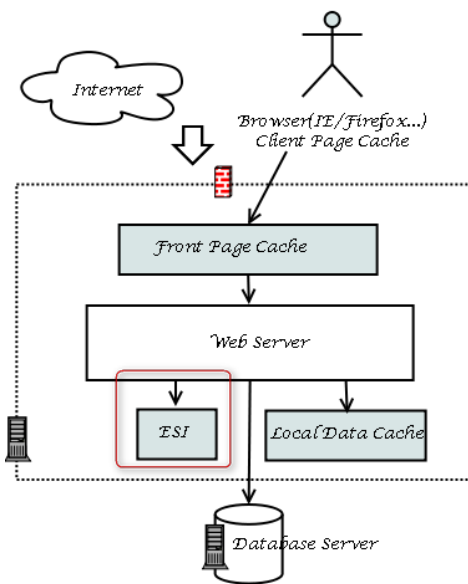


💡 采用具备缓存功能的http反向代理服务器作前端页面缓存器，  
Varnish\Squid\Ncache\AiCache(商业)...【硬件F5】



# 网站架构演变及其技术脉络

## ■[Step2]技术点—页面片段缓存ESI(Edge Side Includes)



芒果网 mango city.com

自助取票 航班风 特价票 环保风 自取行程单 服务全新升级

积分商城 | 代订 | 合作卡 | 芒果知道

青芒果 | 蜜柠檬 | 爱淘

首页 | 国内酒店 | 港澳海外酒店 | 国内机票 | 国际机票 | 旅游度假 | 租车 | 邮轮 | 门票 | 企业客户 | 旅游指南 | 社区

芒果网预订电话: 40066-40066 或 0166-33340066

登录 | 注册 | 我的订单 | 我的账户 | 帮助中心

国内机票查询

航班选择 \* 系统 \* 往返

出发城市 中文/拼音 到达城市 中文/拼音

出发日期 2011-01-08 出发时间 全部

航空公司 全部 舱位等级 全部

查询

最新项目

- 客服热线-机票问答芒果网
- 国航新增广州至广元往返航班
- 国航新增广州至梧州、梧州至...
- 国内航线新增附加费请速通知
- 东航开通北京至三亚直飞航班
- 南航11月新增航班信息
- 国航新增广州至梧州、梧州至...
- 吉祥航空新增上海至宜昌、宜昌...
- 东北航新增航班信息

帮助中心

- 机票预订须知
- 机票预订步骤
- 机票退改服务

主要城市近期特价机票

公司航班信息请点击查看

城市	上海	广州	深圳	成都	南京	武汉	杭州
出发日期	2011-01-07	2011-01-08	2011-01-09	2011-01-10	2011-01-11	2011-01-12	2011-01-13
上海	¥580	¥790	¥680	¥850	¥850	¥790	¥790
深圳	¥1490	¥1750	¥1400	¥1750	¥1750	¥1400	¥1400
成都	¥360	¥360	¥940	¥1010	¥940	¥940	¥940
昆明	¥910	¥1000	¥910	¥1000	¥910	¥910	¥1090
广州	¥1700	¥1700	¥1700	¥1280	¥1280	¥1360	¥1360
杭州	¥460	¥920	¥920	¥920	¥920	¥920	¥920
乌鲁木齐	¥1210	¥1450	¥1690	¥1690	¥1810	¥1930	¥1930
重庆	¥360	¥940	¥940	¥940	¥360	¥940	¥940

以上价格每半小时更新一次

网站地图/意见反馈

关于芒果 | 网站导航 | 广告服务 | 商务合作 | 芒果新闻 | 诚聘英才 | 芒果招聘 | 联系我们 | 友情链接

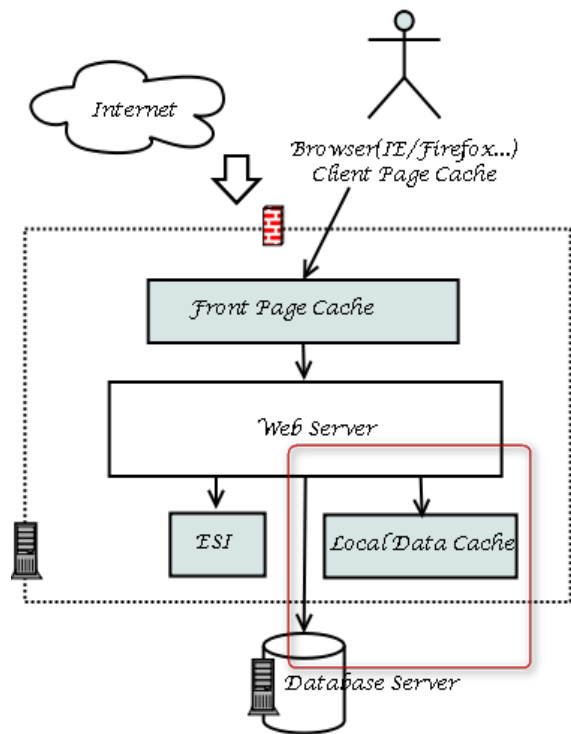
Copyright © 2005-2011, mangocity.com. All Rights Reserved. 芒果网 版权所有



ESI需要服务器端支持，常见apache(mod\_esi)、WebLogic、JSP标签库(JESI)等。

# 网站架构演变及其技术脉络

## ■[Step2]技术点—本地数据缓存



### 技术点说明

关系数据库系统（如：Oracle\MySQL）Query Cache策略：一般以sql为key来缓存查询结果，尽量不要拼sql，使用PreparedStatement的“？”模式sql；Query Cache大小要根据数据库系统具体情况合理设置，过大只会浪费内存，参考值：128M

关系数据库系统Data Buffer策略：就是数据库数据内存缓存器，其访问命中率决定数据库性能，可根据实际物理内存大小适量增大，如：MySQL建议buffer值为物理内存60-80%

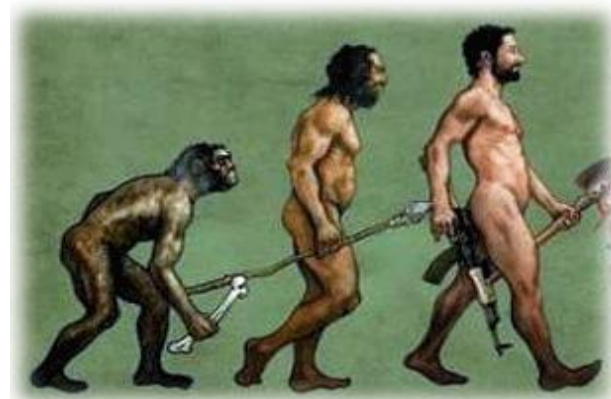
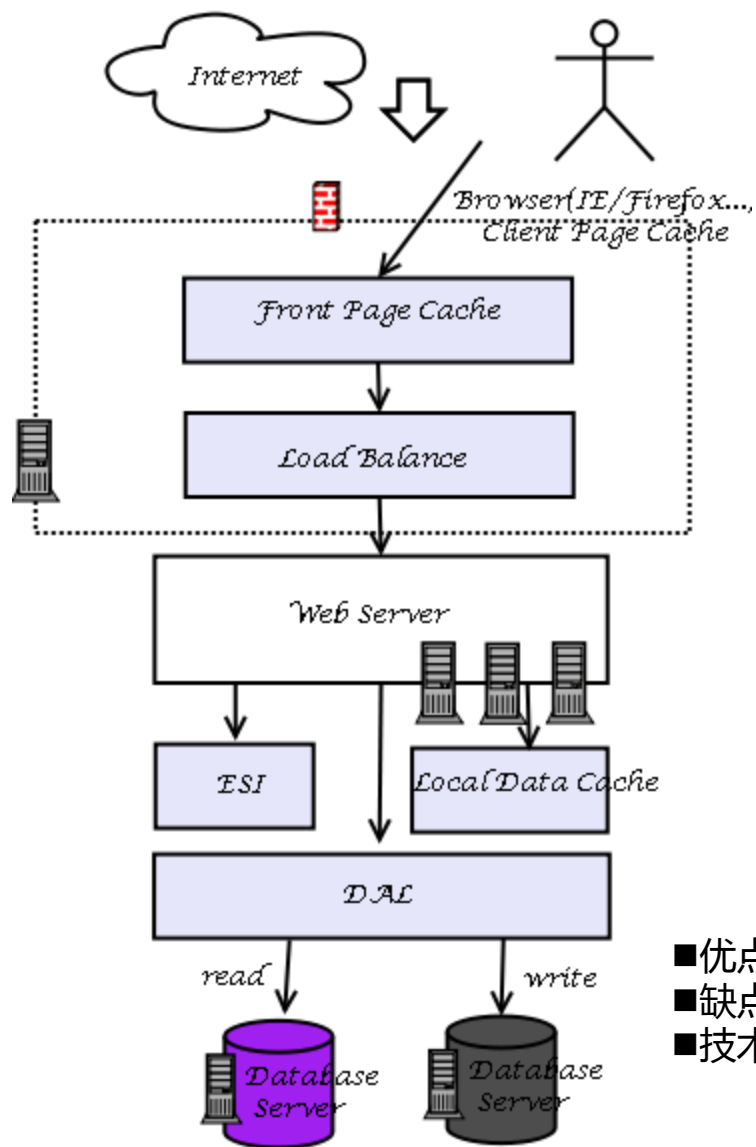
应用服务器Cache包括：对象缓存（例如：对象线程安全，做成单例），更新频率不大数据考虑缓存（如：基表数据、配置文件信息），考虑使用线程池，对象池，连接池等

常见java解决方案：map\OSCache\EHCache等

💡 需要从数据库系统和Web应用服务器两个层面考虑缓存优化

# 网站架构演变及其技术脉络

## ■[Step3]增加机器做HA、数据库读写分离

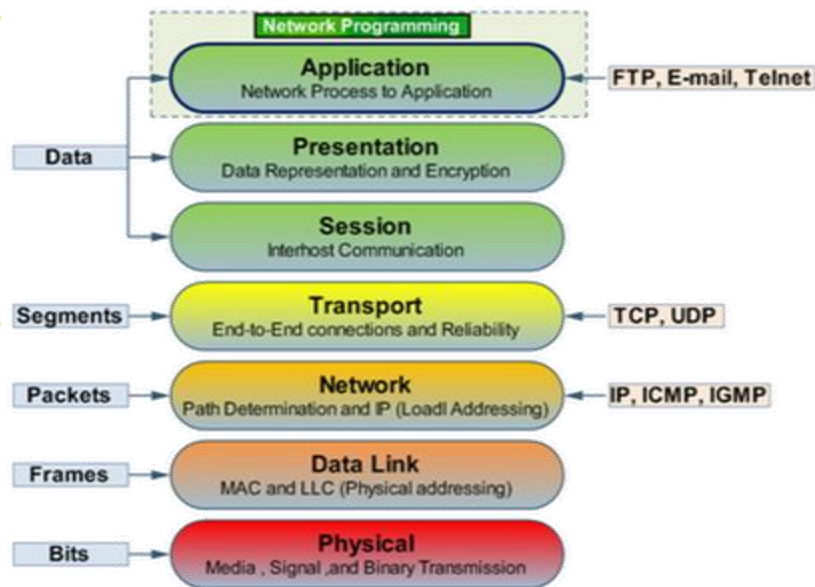


- 优点：增加服务器和HA机制，系统性能及可用性得到保证
- 缺点：读写分离，增加程序难度，架构变复杂，维护难度增加
- 技术点：负载均衡、DAL、数据库读写分离

# 网站架构演变及其技术脉络

## ■[Step3]技术点—负载均衡

### Layer4-7 Switch



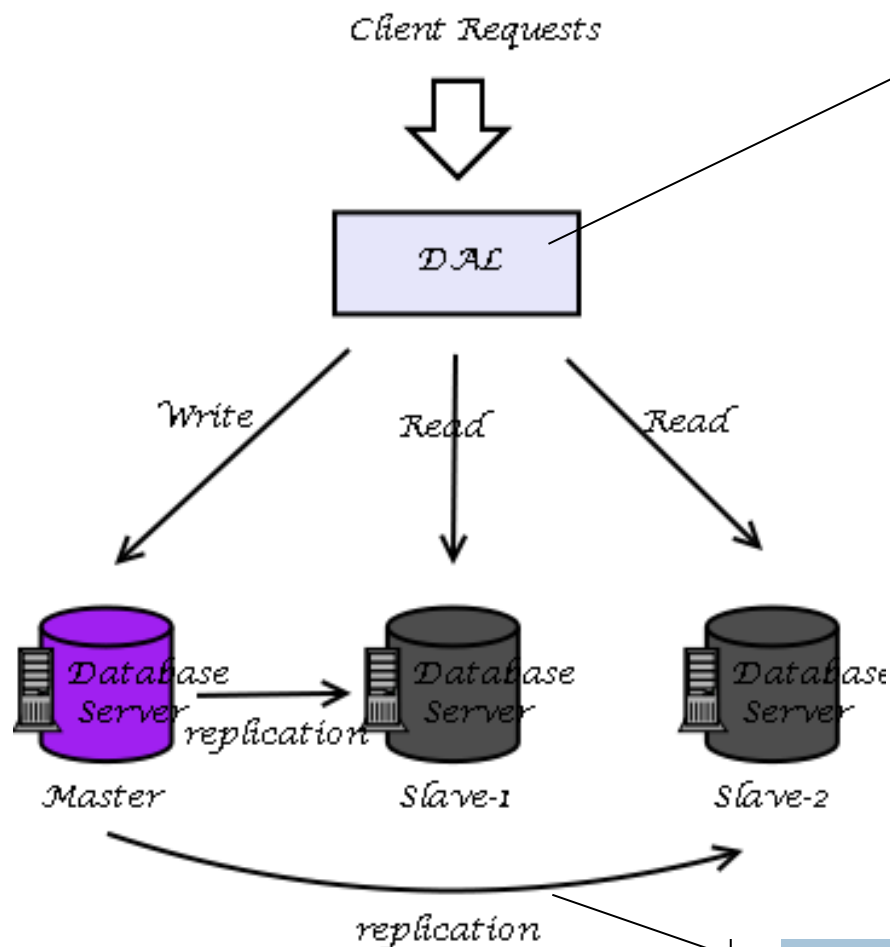
类型	说明
DNS负载均衡	实现简单、有Cache缺乏灵活性，但对分区域（如构建CDN方案）访问简单有效
反向代理软件	HAProxy、Nginx、Apache、Lighttpd等
硬件产品	F5、NetScaler等
LVS(Linux Virtual Server)	<a href="http://www.linuxvirtualserver.org/">http://www.linuxvirtualserver.org/</a>
SMART Client	自己写代码某些情况下简单有效





# 网站架构演变及其技术脉络

## ■[Step3]技术点—数据库读写分离及DAL



- 读写分离逻辑分批
- 负载均衡
- 失效转移 (failover)
- 数据库分区透明支持
- 两大实现模式：独立Proxy服务器；单独API库文件

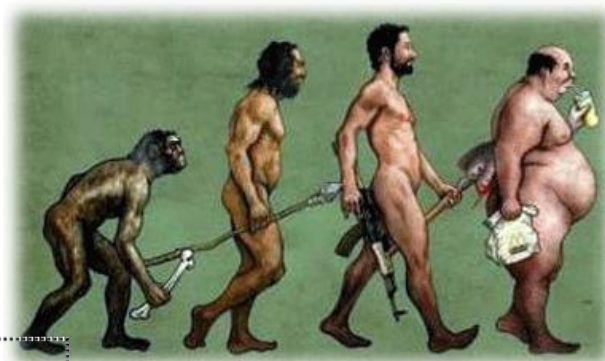
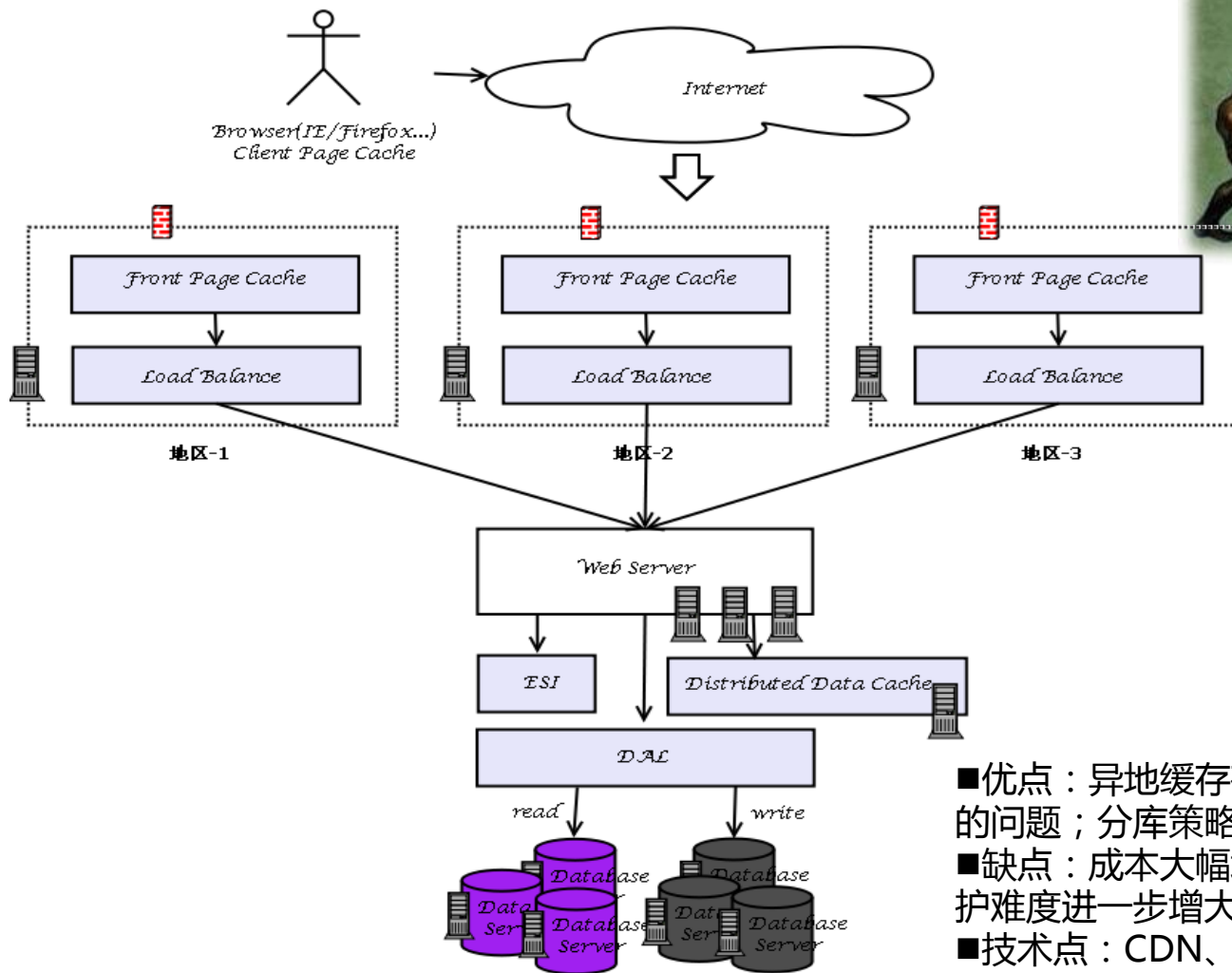
各个数据库厂商都有自己复制方案  
常见通用方案：ETL、GoldenGate  
TJS...





# 网站架构演变及其技术脉络

## ■[Step4]CDN、分布式缓存、分库



- 优点：异地缓存有效解决不同地方用户访问过慢的问题；分库策略带来网站性能整体提升
- 缺点：成本大幅增加，架构进一步复杂化，也维护难度进一步增大，架构开始臃肿了
- 技术点：CDN、分布式缓存、Shard分库

# 网站架构演变及其技术脉络

## ■[Step4]技术点—CDN



■CDN(Content Delivery Network)内容分发网络

■将网站的内容分发到最接近用户的网络“边缘”，使用户可以就近获取，从而解决互联网网络拥挤的状况，提高用户访问的响应速度。

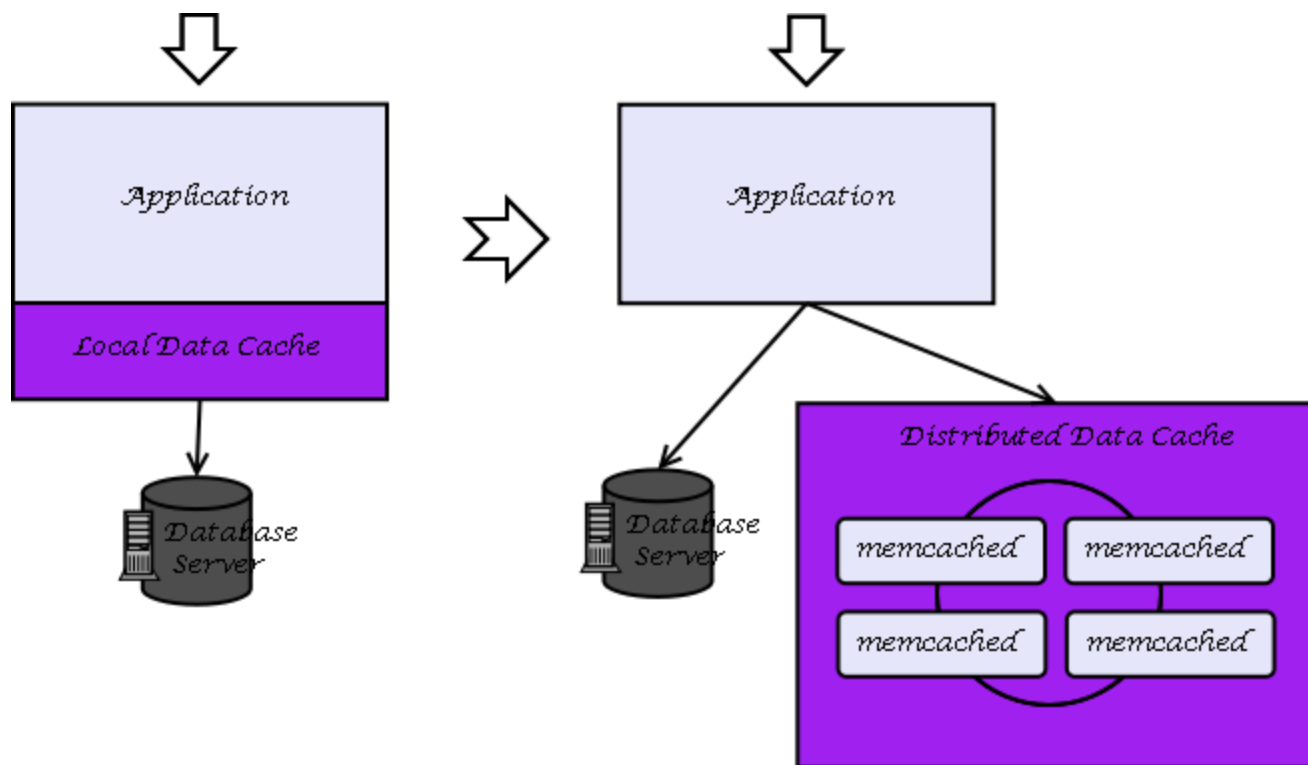
■适合静态内容很多（如：静态页面、图片、视频等）及页面内容实时性要求不高的网站，如：新闻类门户网站

■CDN构建可以做的很简单，也可以很复杂，主要根据自己网站实际情况而定



# 网站架构演变及其技术脉络

## ■[Step4]技术点—分布式缓存



- 本地缓存性能优秀，但容量有限，无伸缩性
- 采用分布式缓存方案突破容量限制，具备良好伸缩性；但分布式涉及远程网络通信消耗其性能本地缓存来得优秀，并可涉及节点状态维护及数据复制问题，其稳定性和可靠性是个挑战。
- 目前流行分布式缓存方案：memcached、membase、redis等，基本上当前的NoSQL方案都可以用来做分布式缓存方案

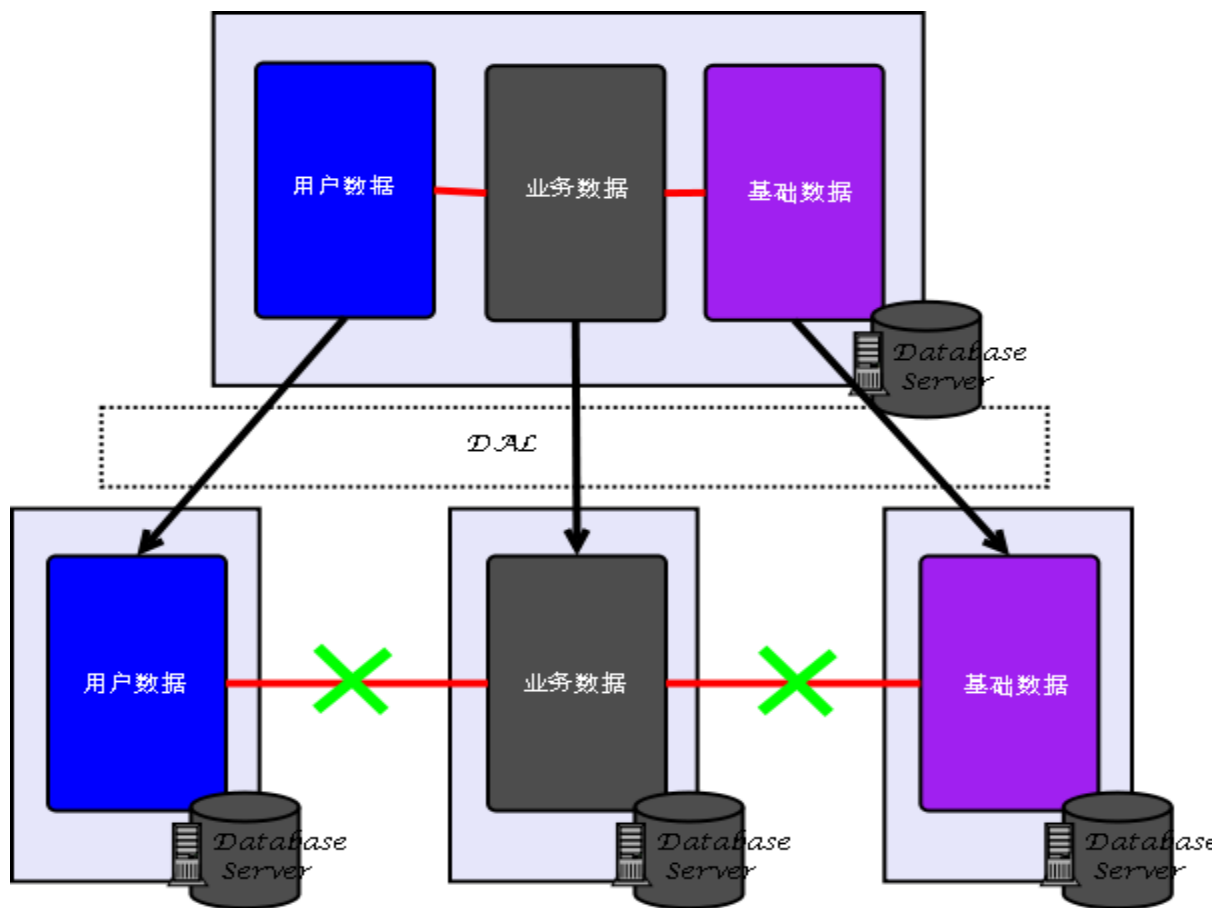


# 网站架构演变及其技术脉络

## ■[Step4]技术点—分库

■读写分离（简单有效，前面已介绍）

■垂直分区

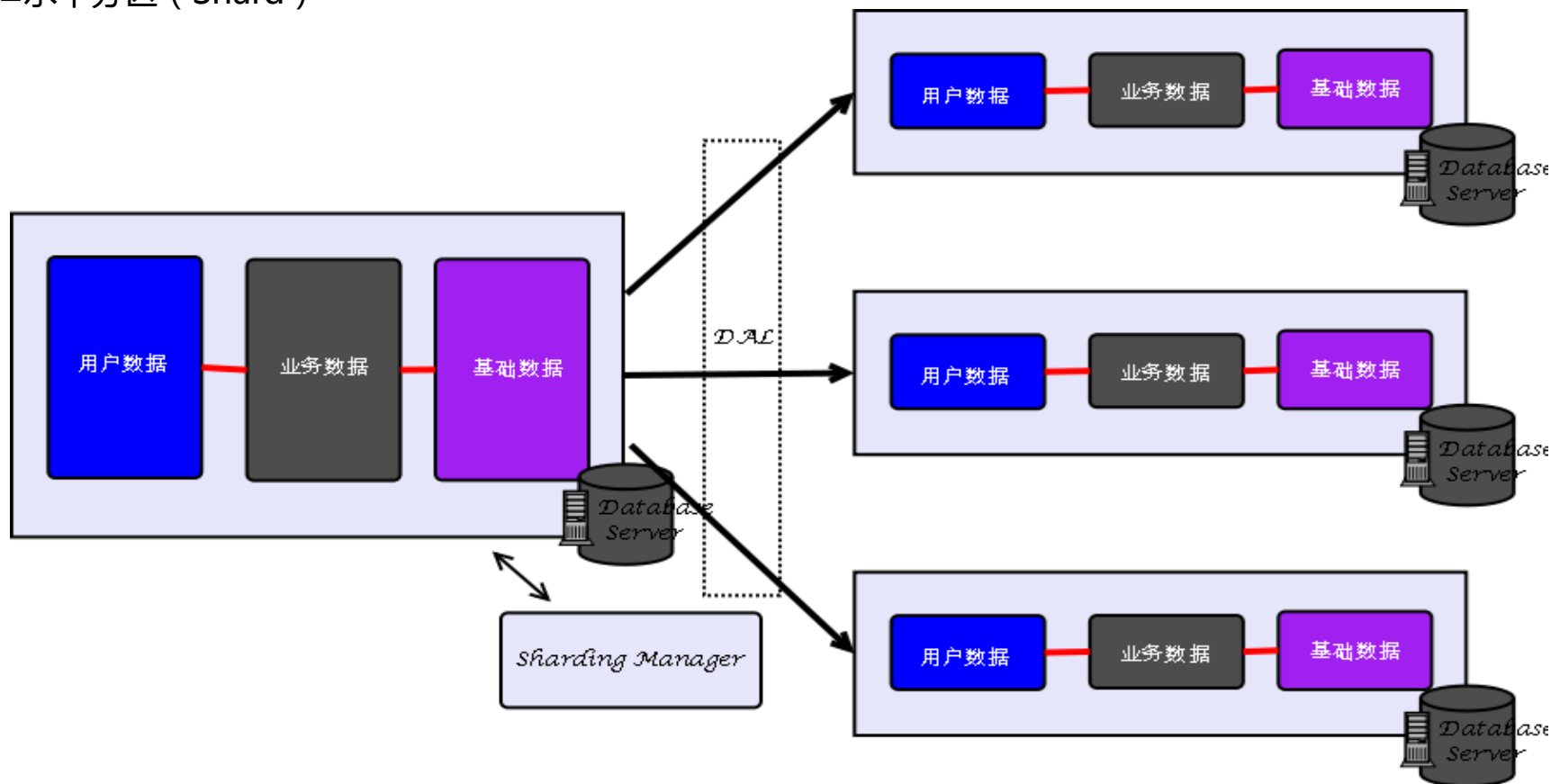


良好的松耦合的模块化设计是垂直分库的前提

# 网站架构演变及其技术脉络

## ■[Step4]技术点—分库

### ■水平分区 (Shard)

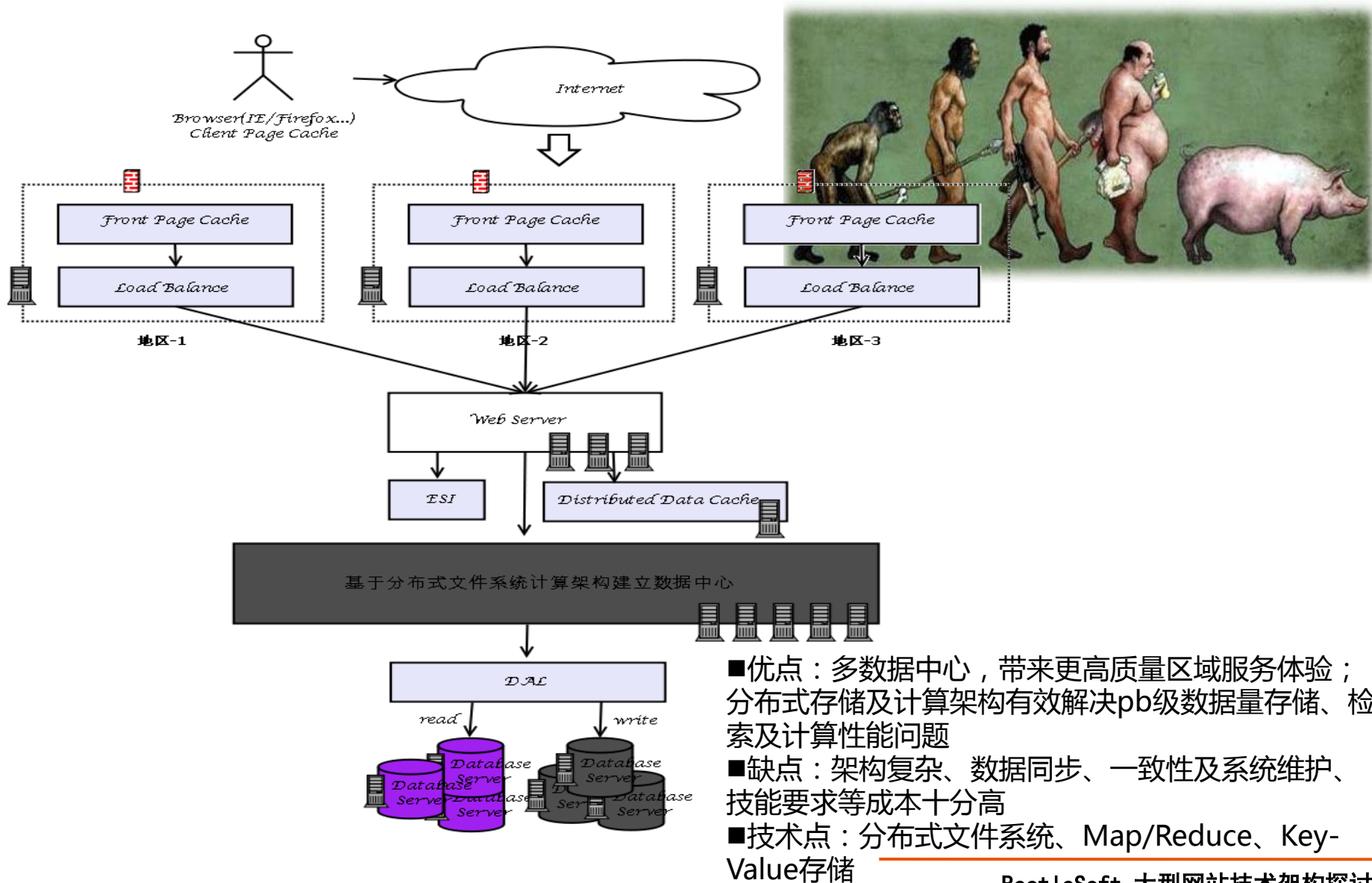


- 💡 分片Key识别（划分检索依据）是关键
- 💡 是否还有其它招？用NoSql数据库部分替换关系数据库



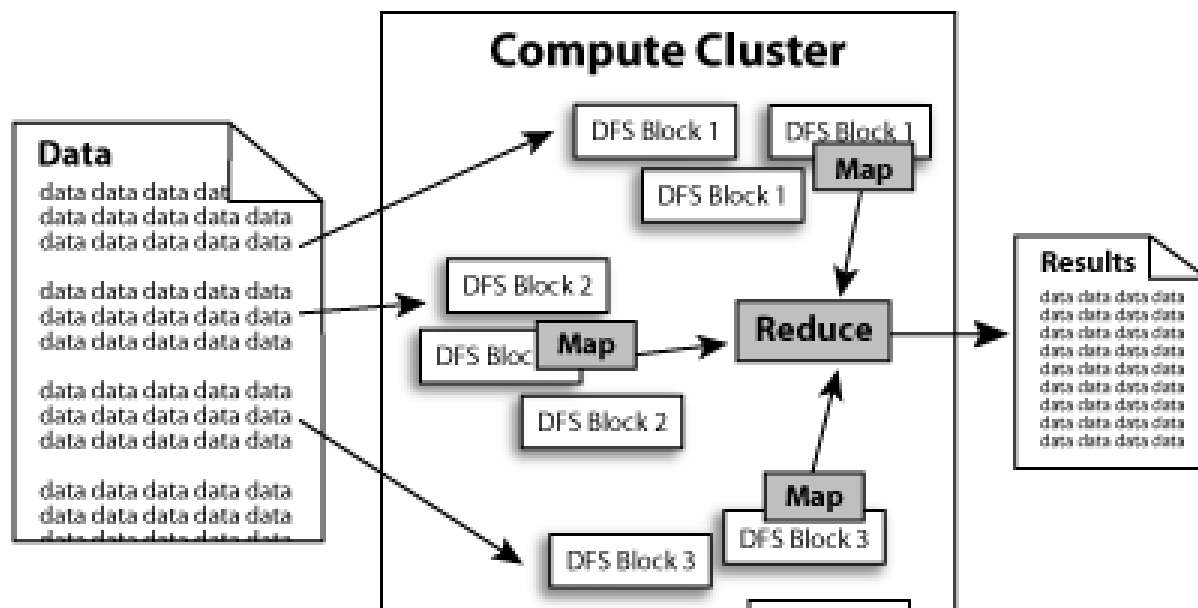
# 网站架构演变及其技术脉络

## ■[Step5]多个数据中心，向分布式存储和计算的架构体系迈进

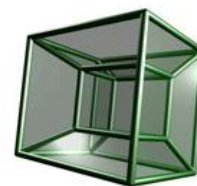


# 网站架构演变及其技术脉络

## ■[Step5]技术点—向分布式存储计算解决方案[DFS、Map/Reduce、Key-Value DB]



- DFS分布式文件系统，如：Lustre\HDFS\GFS\TFS\FreeNas等
- Map/Reduce算法（计算框架），基本上现有NoSQL数据库中都支持此算法。
- Key-Value DB，也作为NoSQL解决方案，如：BigTable\Tair\Hbase\HyperTable等
- 提供完整解决方案：
  - Google(GFS|Map/Reduce|BigTable)
  - Apache Hadoop(HDFS|Map/Reduce|HBase)





**大型网站架构的目标与挑战**

**网站架构演变及其技术脉络**

**架构设计理论与原则**

**讨论及总结**





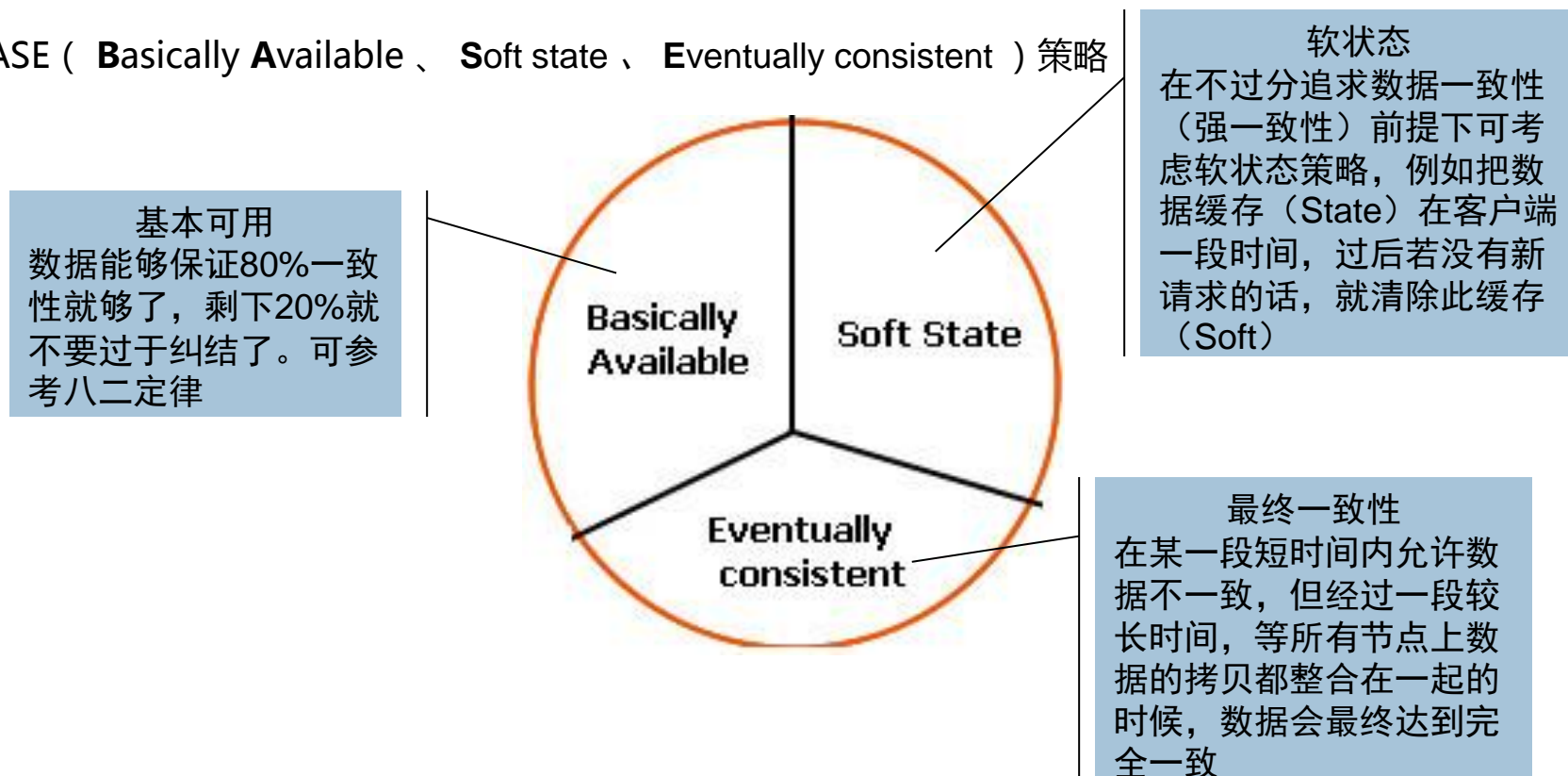
# 架构设计理论与原则

## ■关于数据一致性—ACID vs BASE

■ACID ( Atomicity 、 Consistency 、 Isolation 、 Durability ) 是关系型数据库的最基本原则，遵循ACID原则强调一致性，对成本要求很高，对性能影响很大。

■问题：ACID原则适用于互联网应用吗？可用性似乎比一致性重要些

■BASE ( Basically Available 、 Soft state 、 Eventually consistent ) 策略

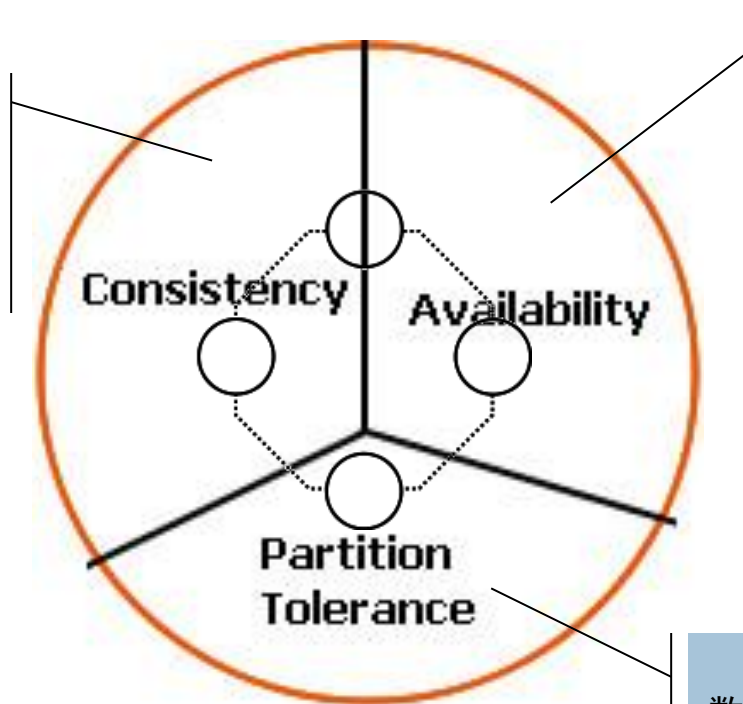


BASE策略与ACID不同，其基本思想就是通过牺牲强一致性，以获得更好的可用性或可靠性

# 架构设计理论与原则

## ■关于分布式系统—CAP理论

**一致性**  
分布式系统中，数据一般会存储在不同节点，一致性就是要保证对数据操作的原子性



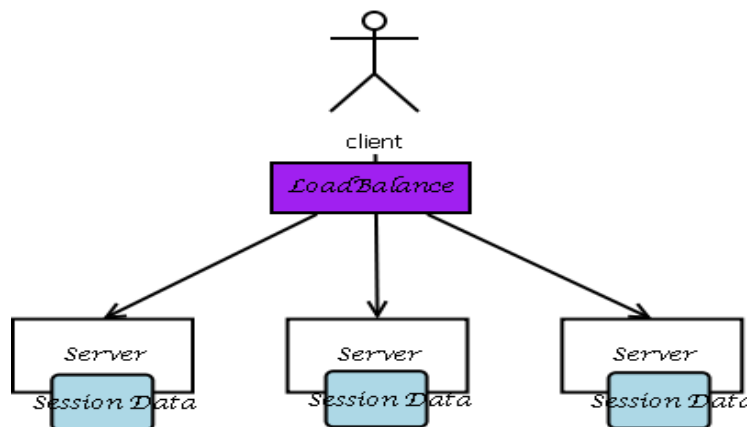
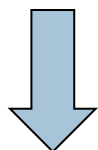
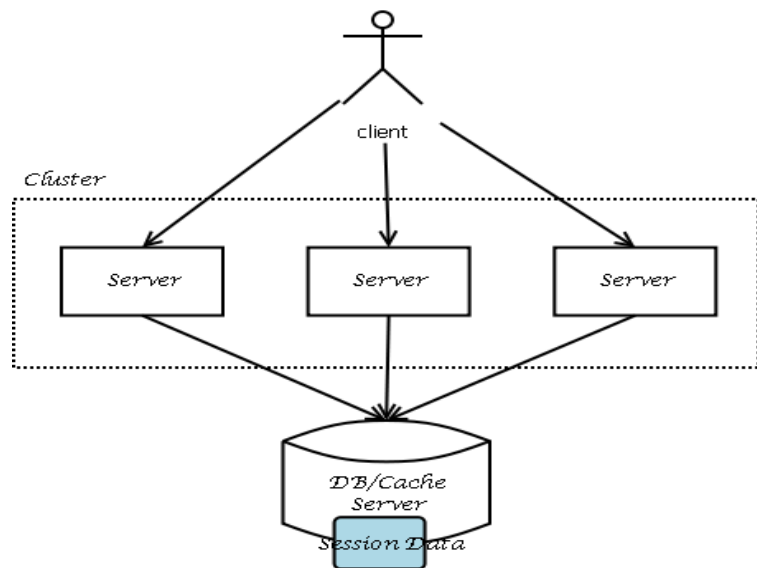
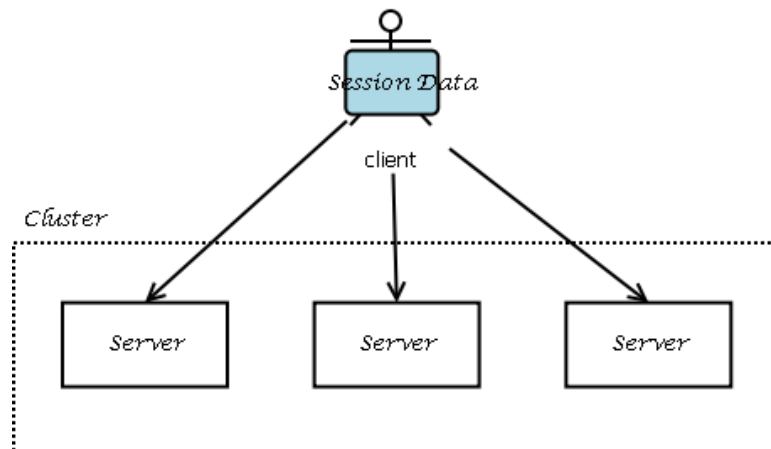
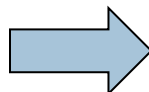
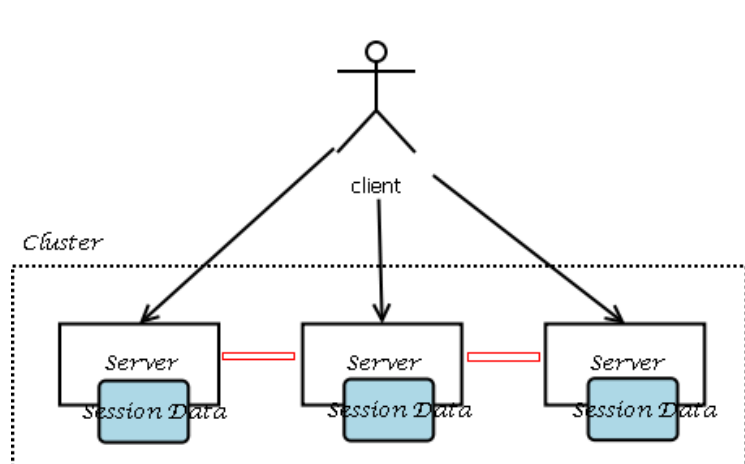
**可用性**  
确保客户访问数据时可得到响应。不强调各个节点上数据要保持一致性。

**分区容忍性**  
数据分区存储后，即使部分分区组件不可用，其施加的操作也能够完成

💡 CAP理论指出：一个分布式系统不可能同时满足一致性、可用性和分区容忍性这三项需求，最多只能同时满足其中两个。

# 架构设计理论与原则

## ■ 无共享架构 ( Share Nothing Architecture )



# 架构设计理论与原则

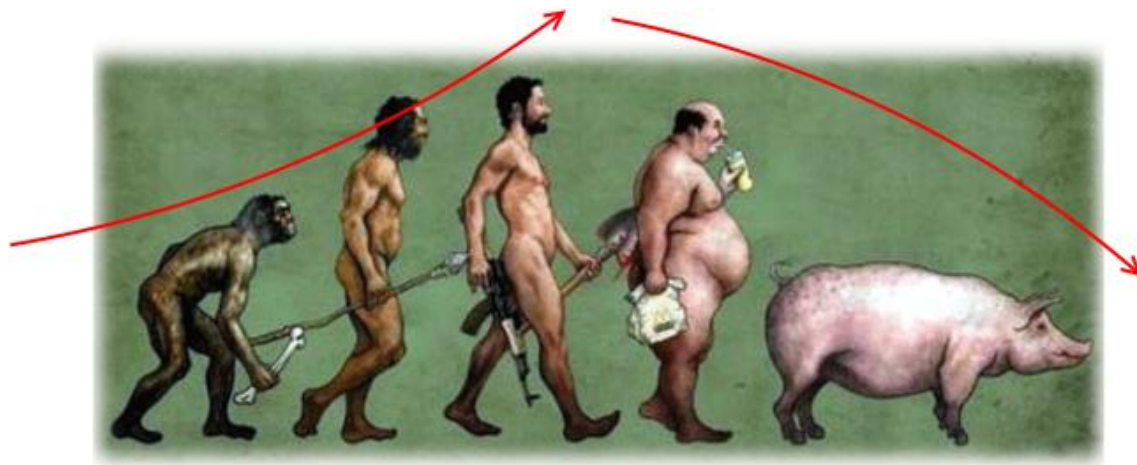
## ■ ED-SOA架构

- ED-SOA，事件驱动，面向服务架构

- SOA是系统组件化、模块化构建性理论；ED是系统组件之间同步通信，采取事件机制异步化，提高响应速度

- 基于ED-SOA构建松耦合系统可以显著改善网站可伸缩性

## ■ 架构进化与退化--奥卡姆剃刀原理

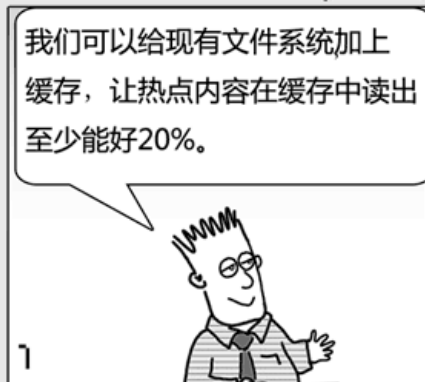
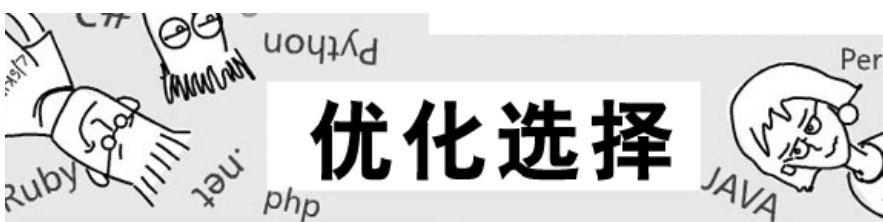


- 进化—寻找最适合的；退化—简化不必要的

- 简单就好，慎防过渡设计

# 架构设计理论与原则

## ■ 考量成本，先硬后软原则



**大型网站架构的目标与挑战**

**网站架构演变及其技术脉络**

**架构设计理论与原则**

**讨论及总结**



# 讨论及总结

- 大型网站架构是什么样子的？
- 存在万能的架构吗？架构本质是什么？



- 网站架构如何选型？开发语言重要吗？

- 架构只是浮云？神马才是重要的？。。。





**Thank you ! - Q&A**