目录

[1 互联网架构演化 2](#_Toc468612236)

[1.1 大型网站特点 2](#_Toc468612237)

[1.1.1 高并性、大流量 2](#_Toc468612238)

[1.1.2 高可用性 2](#_Toc468612239)

[1.1.3 海量数据 2](#_Toc468612240)

[1.1.4 用户分布广泛性、网络情况复杂 3](#_Toc468612241)

[1.1.5 安全环境恶劣 3](#_Toc468612242)

[1.1.6 需求快速变更、发布频繁 3](#_Toc468612243)

[1.1.7 渐近式发展 3](#_Toc468612244)

[1.2 大型网站演化历程 3](#_Toc468612245)

[1.2.1 初始阶段网站演化历程 3](#_Toc468612246)

[1.2.2 应用服务和数据服务分离演化历程 4](#_Toc468612247)

[1.2.3 使用缓存提升查询效率演化历程 4](#_Toc468612248)

[1.2.4 使用缓应用服务器集群改善网站并发处理演化历程 5](#_Toc468612249)

[1.2.5 数据库读写分离演化历程 6](#_Toc468612250)

[1.2.6 反向代理和CDN加速网站响应演化历程 7](#_Toc468612251)

[1.2.7 分布式文件系统及分布式数据库系统演化历程 7](#_Toc468612252)

[1.2.8 NOSQL和搜索引擎系统演化历程 8](#_Toc468612253)

[1.2.9 业务拆分网站演化历程 9](#_Toc468612254)

[1.2.10 分布式服务演化历程 10](#_Toc468612255)

[1.3 大型网站演化的意义 11](#_Toc468612256)

[1.3.1 大型网站演化的核心意义是需求灵活多变性 11](#_Toc468612257)

[1.3.2 大型网站发展演化主要取决于业务的日新月异的变化 11](#_Toc468612258)

[1.4 大型网站架构设计的设计误区 11](#_Toc468612259)

[1.4.1 一味照搬网上大公司的解决方案 11](#_Toc468612260)

[1.4.2 太过强调最先进的技术，而忽视业务需求、时间、人员能力及开发成本等因素 12](#_Toc468612261)

[1.4.3 企业用技术解决业务需求或者管理上轻而易举就能解决的问题 12](#_Toc468612262)

[1.5 大型网站架构的核心设计要素 12](#_Toc468612263)

[1.5.1 高性能 12](#_Toc468612264)

[1.5.2 可用性 12](#_Toc468612265)

[1.5.3 伸缩性 12](#_Toc468612266)

[1.5.4 扩展性 13](#_Toc468612267)

[1.5.5 安全性 13](#_Toc468612268)

[1.6 架构师的职责 13](#_Toc468612269)

[1.6.1 基本工作目标 13](#_Toc468612270)

[1.6.2 主要职责 13](#_Toc468612271)

[1.6.3 主要要求 14](#_Toc468612272)

[1.6.4 实践基础 14](#_Toc468612273)

[1.7 架构师成长课程 15](#_Toc468612274)

[1.7.1 Java开发技术等相关课程 15](#_Toc468612275)

[1.7.2 Linux环境及其软件等相关课程 15](#_Toc468612276)

[1.7.3 负载均衡应用服务器等相关课程 16](#_Toc468612277)

[1.7.4 Web应用服务器等相关课程 16](#_Toc468612278)

[1.7.5 缓存应用服务器等相关课程 16](#_Toc468612279)

[1.7.6 数据库应用服务器等相关课程 16](#_Toc468612280)

[1.7.7 消息中间件应用技术等相关课程 16](#_Toc468612281)

[1.7.8 搜索引擎应用技术等相关课程 17](#_Toc468612282)

[1.7.9 NOSQL等数据库相关课程 17](#_Toc468612283)

[1.7.10 大数据应用技术等相关课程 17](#_Toc468612284)

[1.7.11 持续部署应用技术等相关课程 17](#_Toc468612285)

[1.7.12 运维监控技术等相关课程 17](#_Toc468612286)

# 互联网架构演化

## 大型网站特点

### 高并性、大流量

大型网站面对高并发用户大流量访问。Google日均PV（**page view，页面浏览量）**数35亿，日均IP访问数3亿。腾讯QQ 2016年最大在线人数2.59亿；支付宝支付峰值12万笔/秒；

2016年天猫淘宝成交额1207亿，活动一开始，第一分钟内独立访问用户达到1000多万。

这么大的高并发性及大流量访问，对网站的架构设计无疑是一种巨大的挑战。

### 高可用性

系统一般要求是7\*24小时不间断提供服务，否则一旦服务中断，将会带来巨大的损失和影响。例如，[2015 年 5 月 28 日的携程网宕机事件](http://www.baidu.com/link?url=ru9q_DyypUZsW6I-5E25Db8d7m7h-zP0o802bOQ0Ql5A4WlOR9Hq-kdM3pxvbj9h&wd=&eqid=ef80954300031b27000000035843540c" \t "_blank)，全网业务中断12小时。5月29日，携程发布官方情况说明称，此次事件是由于员工错误操作，删除了生产服务器上的执行代码导致。携程网站宕机还影响到了在线旅游市场供应商及其他合作伙伴。同程网就表示，因为部分酒店订单来自携程，导致其网上给的部分酒店订单无法成交。但是同程网对携程表示理解和支持。携程的合作伙伴阳光车导还发声表示攻击携程的敌人也是自己的敌人。引用来自<http://www.yunweipai.com/archives/4978.html>

### 海量数据

需要存储和管理海量数据，需要使用大量服务器。例如,facebook每周照片数目接近10亿。百度收录网页数目有数百亿，Google有近百万台服务器为全球用户提供服务。

### 用户分布广泛性、网络情况复杂

许多大型网站都是为全球用户提供服务的，用户分布范围较广，各地网络情况千差万别。在国内，各个运营商网线互通难的问题，往往导致网络访问会出现网速缓慢或者有时不能使用等现象。另外有时天灾人祸，例如某个地方海底发生地震，造成海底光缆数次故障，也让一些对国外用户依赖较大的网站不得不考虑在海外建立数据中心。

### 安全环境恶劣

由于网站的开放性，使得互联网站更易受到来自四面八方的黑客攻击。例如，2011年12月21日，黑客在网上公开了知名程序员网站CSDN的用户数据库，高达600多万个明文的注册邮箱账号和密码遭到曝光和外泄，成为中国互联网历史上一次具有深远意义的网络安全事故。引用来自http://tech.qq.com/a/20111221/000485.htm

### 需求快速变更、发布频繁

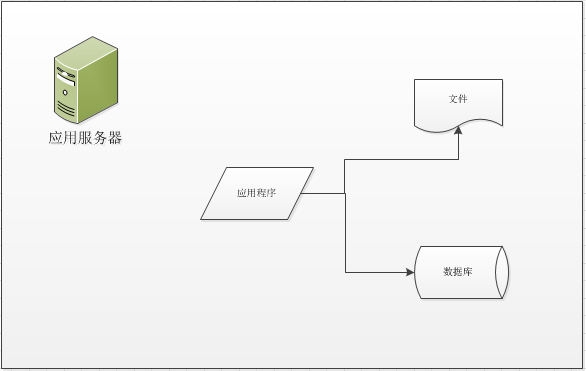
和传统的软件版本发布频率不同，互联网产品为快速适应市场，满足用户需求，其产品发布频率是极高的。例如，传统的office办公软件以年为单位发布的，而一般大网站的产品每周都可能有新版本上线，至于中小网站的发布就更频繁了，有时一天发布都有好几次。

### 渐近式发展

互联网站，可以说是一个迭代产品。每个版本都是基于上一个版本的基础上修改而成。因此，比较适合用目前比较流行的敏捷开发模式来开发。

## 大型网站演化历程

### 初始阶段网站演化历程



### 应用服务和数据服务分离演化历程



### 使用缓存提升查询效率演化历程



### 使用缓应用服务器集群改善网站并发处理演化历程



### 数据库读写分离演化历程



### 反向代理和CDN加速网站响应演化历程



### 分布式文件系统及分布式数据库系统演化历程



### NOSQL和搜索引擎系统演化历程



### 业务拆分网站演化历程



### 分布式服务演化历程



## 大型网站演化的意义

### 大型网站演化的核心意义是需求灵活多变性

一个大型网站都是通过最初的小型网站业务发展，然后随着每天灵活多变的业务需求，通过很长的技术演化过程中，不断的解决各种各样的问题发展壮大起来的。

### 大型网站发展演化主要取决于业务的日新月异的变化

可以说，技术是为业务需求服务的。脱离了业务再谈技术，有时往往走入误区。

## 大型网站架构设计的设计误区

### 一味照搬网上大公司的解决方案

由于大公司巨大成功的光环效应，再加上一些大公司技术高手的影响，在作网站决策时，不切实际，容易盲从，导致架构与业务场景不相符合，从而造成一开始就错了。

### 太过强调最先进的技术，而忽视业务需求、时间、人员能力及开发成本等因素

在选型过程中，一昧追求技术的先进性，感觉一个项目，能用的先进技术全给他用上，与人谈起来，倍儿有面子，但有时会导致选来选去，无所适从，浪费了大量的时间和精力，有时还会忽视人员能力与开发成本，从而导致架构进入了一个误区.

### 企业用技术解决业务需求或者管理上轻而易举就能解决的问题

能用技术解决的问题，尽量用技术，但有时业务上能解决大量技术上不必要解决的问题，此时，就不应该钻牛角尖用技术解决。

## 大型网站架构的核心设计要素

### 高性能

1. 浏览器端，浏览器缓存、使用页面压缩、合理布局、减少cookie传输等手段
2. CDN：将网站静态内容分发至离用户最近的服务商机房，使用最短路径获取数据，也可以部署反向代理服务器，缓存热点文件，加快请求响应速度，减轻应用服务器负载压力。
3. 应用服务器：可以使用服务器本地缓存和分布式缓存，通过缓存在内存中的热点数据处理用户请求，加快请求处理过程，减轻数据库负载压力。
4. 代码层：使用多线程，改善内存管理等手段优化性能。
5. 数据库服务器：索引、缓存、SQL优化等

### 可用性

1. 应用服务器宕机，可以采用集群负载均衡来保证服务的正常运行。
2. 数据库服务器的宕机，要保证对数据的实时备份。如常用的Master-Slave主从，Master-Master主-主模式。

### 伸缩性

1. 应用服务器集群保证系统的伸缩性，即不可能只有一台应用服务器提供服务。
2. 缓存服务器集群保证缓存端的可伸缩性，即不可能只有一台缓存应用服务器提供缓存服务
3. 数据库应用服务器：如主-从备份，但很难做到大规模集群的可伸缩性，因此，通过路由分区等手段将多个数据库的服务器组成一个集群，如Nosql分区，可以应对海量数据存储。

### 扩展性

1. 采用事件驱动架构来提供可扩展性，即利用消息中间件技术（如jms、kafka、activemq等）来实现。
2. 采用分布式服务来实现将业务和可复用服务分离开来，然后分布式服务框架来实现调用。常用的有restful服务。

### 安全性

1. **加强空间或服务器的安全**
2. **加强内部数据的管理及监督**
3. **检查站点程序是否存在漏洞**
4. **整治常用电脑，杜绝木马病毒**
5. 采用安全性的https协议。
6. 尽量采用不可逆的MD5算法来对数据库中的用户密码进行加密处理。
7. 尽量避免使用root等默认用户。
8. 对重要的数据进行加解密处理。
9. 其它安全性手段。如对已经暴露的安全漏洞要及时升级等。

## 架构师的职责

### 基本工作目标

1.确保公司软件研发工作的目标与公司产品发展规划及公司长期远景目标相一致

2.确保公司各类项目的技术路线符合公司整体要求与规范。

3.确保个项目的技术选型、技术架构设计。

4.确保技术架构理念传导到设计人员与开发人员。

### 主要职责

1.负责理解和管理非功能性系统需求，包括软件的可维护性、性能、复用性、可靠性、有

效性和可测试性等。

2.负责组织技术研究和攻关工作，组织及带领公司内部员工研究与项目相关的新技术。

3.协助项目经理制定项目计划和控制项目进度。

4.根据产品部所提出的的需求，对开发团队所提出的设计进行技术层面的把关。

5.协助产品部完成《用户需求说明书》、《需求变更说明书》。

6.负责对整个软件架构、关键构件、接口的设计。协助设计人员完成《系统概要设计说明

书》

7.负责软件测试、集成、交付等过程中所需的接口规范和技术支持。

### 主要要求

1.具备6年以上软件行业工作经验；

2.具备4年以上C/S体系结构软件产品开发及架构和设计经验；

3.具备3年以上的代码编写工作经验；

4.具备丰富的大中型开发项目的总体规划、方案设计及技术队伍管理经验；

5.对相关的技术标准有深刻的认识，对软件工程标准规范有良好的把握；

6.对JAVA技术及整个解决方案有深刻的理解及熟练的应用，并且精通WebService/j2ee架构和设计模式，并在此基础上设计产品框架；

7.具有面向对象分析、设计、开发能力（OOA\OOD\OOP），精通UML和ROSE，熟练使用Rational Rose、PowerDesigner等工具进行设计开发；

8.精通大型数据库如MySQL、Oracle、Sql Server等的开发；

9.对计算机系统、网络和安全、应用系统架构等有全面的认识，熟悉项目管理理论，并有实际运维等操作能力。

### 实践基础

1.在应用系统开发平台和项目管理上有深厚的基础，有大中型应用系统开发和实施的成功案例；

2.良好的团队意识和协作精神，有较强的内外沟通能力。

## 架构师成长课程

### Java开发技术等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | Spring IOC | Spring IOC的基本概念、原理与使用 |
|  | Spring AOP | Spring AOP的基本概念、原理与使用 |
|  | Spring事务 | Spring事务的基本概念、原理与使用 |
|  | SpringMVC | SpringMVC的基本概念、原理与使用 |
|  | JAVA设计模式 | JAVA设计模式的基本概念、原理与使用 |
|  | Webservice及cxf技术 | Webservice及cxf技术的基本概念、原理与使用 |
|  | hessian技术 | hessian技术的基本概念、原理与使用 |
|  | I/O,NIO技术 | I/O,NIO技术的基本概念、原理与使用 |
|  | Netty技术 | Netty技术的基本概念、原理与使用 |
|  | velocity技术 | velocity技术的基本概念、原理与使用 |
|  | Freemark技术 | Freemark技术的基本概念、原理与使用 |
|  | 多线程技术 | 多线程技术的基本概念、原理与使用 |
|  | Quartz定时任务技术 | Quartz定时任务技术的基本概念、原理与使用 |
|  | Ant | Ant的基本概念、原理与使用 |
|  | Maven | Maven的基本概念、原理与使用 |
|  | Node.js | Node.js的基本概念、原理与使用 |

### Linux环境及其软件等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | Linux基本命令及操作 | 介绍如何用vmware搭建linux环境、linux基本命令及其相关操作等内容 |
|  | Linux shell脚本 | 介绍如何linux shell脚本在实际项目中的使用。 |
|  | LVS+keepalived | 介绍如何linux中如何搭建LVS+keepalived集群环境。 |
|  | rsync+inotify技术 | 介绍如何linux集群环境中如何用rsync+inotify同步文件。 |

### 负载均衡应用服务器等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | nginx | 介绍如何用nginx作负载均衡和反向代理服服务器。 |
|  | apache | 介绍如何用apache作负载均衡和反向代理服服务器。 |

### Web应用服务器等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | Tomcat | 介绍如何用tomcat作WeB服务器、tomcat的优化与配置。 |
|  | Jetty | 介绍如何用jetty作轻量级的Web服务器、jetty的优化与配置。 |
|  | Weblogic | 介绍如何用weblogic作Web服务器、weblogic的优化与配置。 |
|  | JBOSS | 介绍如何用jboss作Web服务器、jboss的优化与配置。 |

### 缓存应用服务器等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | Redis | 介绍如何搭建linux下redis缓存服务器环境及如何使用redis作缓存。 |
|  | Memcached | 介绍如何搭建linux下memcached缓存服务器环境及如何使用memcached作缓存。 |

### 数据库应用服务器等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | mysql | 介绍mysql的使用、mysql主-主、主-从原理等常用架构的搭建与使用。 |
|  | Oracle | 介绍oracle的使用、oracle RAC原理及其搭建与使用。 |
|  | SqlServer | 介绍SQLServer的使用、SQLServer集群原理及其搭建与使用。 |

### 消息中间件应用技术等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | kafka技术 | 介绍kafka原理、集群与使用。 |
|  | acivemq技术 | 介绍activemq原理、集群与使用。 |

### 搜索引擎应用技术等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | Solor搜索技术 | 介绍solor搜索原理、集群与使用。 |
|  | ElasticSearch搜索技术 | 介绍ElasticSearch搜索原理、集群与使用。 |

### NOSQL等数据库相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | mongodb | 介绍mongodb基本概念、原理、集群与使用。 |
|  | hbase | 介绍hbase基本概念、原理、集群与使用。 |
|  | cassandra | 介绍cassandra基本概念、原理、集群与使用。 |

### 大数据应用技术等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | hadoop | 介绍hadoop基本概念、原理与使用。 |
|  | zookeeper | 介绍zookeeper基本概念、原理与使用。 |
|  | spark | 介绍spark基本概念、原理与使用。 |
|  | storm | 介绍storm基本概念、原理与使用。 |
|  | flume | 介绍flume基本概念、原理与使用。 |

### 持续部署应用技术等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | jenkins | 介绍jenkins基本概念、原理及如何用jenkins进行可持续集成部署。 |

### 运维监控技术等相关课程

|  |  |  |
| --- | --- | --- |
| **序号** | **课程名称** | **课程内容** |
|  | Zabbix | 介绍zabbix基本概念、原理及如何用zabbix进行服务器监控与使用。 |
|  | Nagios | 介绍Nagios基本概念、原理及如何用Nagios进行服务器监控。 |