大型互联网架构概述

阅读目录

本文旨在简单介绍大型互联网的架构和核心组件实现原理。理论上讲，从安装配置，最佳实践以及源码来剖析各个组件，这个自然是极好的。由于笔者时间以及知识有限，有很多知识没有在工作中亲自实践的机会。所以有些地方语焉不详，还请大家多多指教。

大型互联网架构

解决问题的通用思路是将分而治之(divide-and-conquer)，将大问题分为若干个小问题各个击破。在大型互联网的架构实践中，无一不体现这种思想。

架构目标

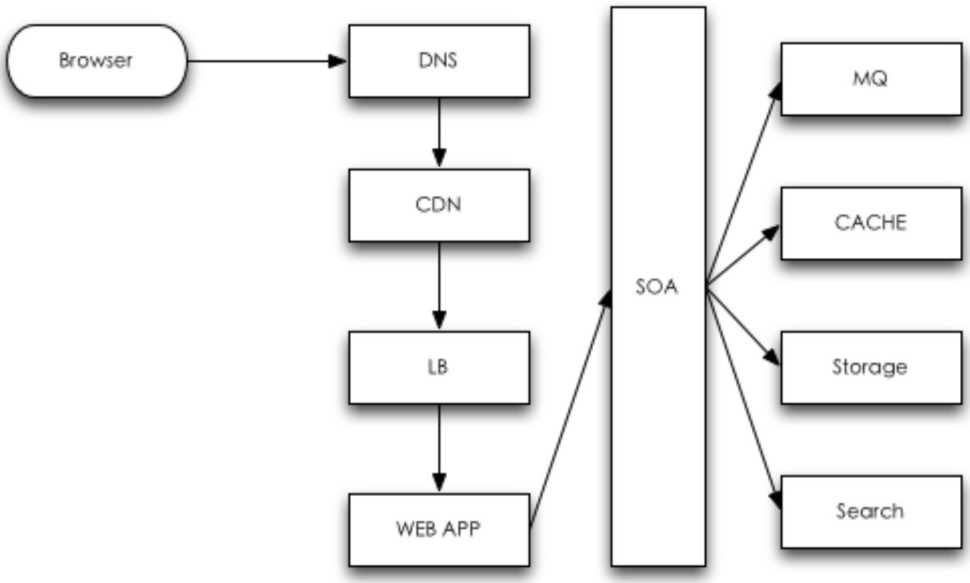
- 低成本:任何公司存在的价值都是为了获取商业利益。在可能的情况下，希望一切都是低成本的。
- 高性能:网站性能是客观的指标，可以具体体现到响应时间、吞吐量等技术指标。系统的响应延迟，指系统完成某一功能需要使用的时间;系统的吞吐量，指系统在某一时间可以处理的数

据总量，通常可以用系统每秒处理的总的数据量来衡量；系统的并发能力，指系统可以同时完成某一功能的能力，通常也用 QPS(query per second)来衡量。
- 高可用：系统的可用性(availability)指系统在面对各种异常时可以正确提供服务的能力。系统的可用性可
以用系统停服务的时间与正常服务的时间的比例来衡量，也可以用某功能的失败次数与成功次数的比例来衡量。
- 易伸缩：注重线性扩展，是否可以容易通过加入机器来处理不断上升的用户访问压力。系统的伸缩性(scalability)指分布式系统通过扩展集群机器规模提高系统性能（吞吐、延迟、并发）、存储容量、计算能力的特性。
- 高安全：现在商业环境中，经常出现被网站被拖库，用户账户被盗等现象。网站的安全性不言而喻。

回到顶部

典型实现

下面典型的一次web交互请求示意图。



回到顶部

DNS

1. 当用户在浏览器中输入网站地址后，浏览器会检查浏览器缓存中是否存在对应域名的解析结果。如果有，则解析过程结束；否则进入下一个步骤
2. 浏览器查找操作系统缓存中是否存在这个域名的解析结果。这个缓存的内容就是操作系统的hosts文件。如果有，则解析过程结束；否则进入下一个步骤
3. 前两个步骤都是本地查找，没有发生网络交互。在本步骤中，会使用到在网络配置的中DNS地址。这个地址我们通常称之为LDNS（Local DNS）。操作系统会把域名发送给LDNS解析。如果解析成功，则解析过程结束；否则进入下一个步骤
4. LDNS将请求返回给GTLD（Global Top Level Domain）服务器，GTLD服务器查找此域名对应的Name Server域名的地址。这个Name Server通常就是你的域名提供商的服务器。Name Server根据客户请求，返回该域名对应的IP地址和TTL(Time To Live)值。
5. 浏览器根据TTL值，把这个域名对应的IP缓存在本地系统中。域名至此解析结束。

回到顶部

CDN

CDN(Content Delivery Network，内容分发网络)部署在网络提供商的机房里面。在用户请求网站服务时，可以从距离自己最近的网络提供商获取数据。比如视频网站和内容网站的热点内容。

如果需要自己搭建CDN系统，有3种主流方案可以选择:

- squid是缓存服务器科班出生,自己实现了一套内存页/磁盘页的管理系统
- varnish是觉得squid性能不行，varnish觉得linux内核已经把虚拟内存管理做得很好了，squid的多此一举反而影响了性能。
- nginx cache是属于不务正业，得益于nginx强大的插件机制。

回到顶部

LB

LB(Load Balance，负载均衡)就是将负载(用户的请求)根据某些策略，将负载分摊给多个操作单元执行。该技术可以提供服务器的响应速度以及利用效率，避免出现单点失效。

这里回顾下前面介绍的两个小节，其实本质上把数据分类(根据数据更新频率，分为动态文件，静态文件)，并把数据放在离距离用户最近的地方。另外一点就是，在DNS和CDN具体实现时，也是大量使用了负载均衡技术。

常见的负载均衡算法由:RR(Round Robin，轮询),WRR(Weighted RR，加权轮询)，Random(随机)，LC(Least Connection，最少连接)，SH(Source Hash,源址哈希)

在常见的互联网架构中，通常使用软件负载:LVS+HAproxy+WebServer(Nginx)。在部署时，LVS，HAProxy，WebServer都会部署一个集群，用来进行负载均衡。LVS工作在第4层，在网络层利用IP地址进行转发。HAProxy工作在第7层，根据用户的HTTP请求(比如根据URL，消息头)来进行转发。

在上述实现中，通常还会使用Keepalived+VIP(虚IP) 技术。Keepalived 提供健康检查，故障转移，提高系统的可用性。通过VIP(配置DNS 绑定域名)的形式对网站进行访问。

回到顶部

WEB APP

前端技术:遵循基本的Web前端优化经验，详见Web前端优化最佳实践及工具集锦介绍。另外还可以使用BigPipe，动态页面静态化，无限滚动的翻页技术等技术提供更好的用户体验。另外一部分就是考虑mobile技术了，这块笔者暂时还没涉足，就不谈了。

后端技术:

- HTTP协议:HTTP协议大概分为请求头，请求体，响应头，响应体。无论是WebServer还是ApplicationServer，很多花样都是基于请求头的请求路径来玩的。
- API接口:使用RESTFUL API，暴露接口。它具有如下好处:1.充分利用 HTTP 协议本身语义。2.面向资源，一目了然，具有自解释性。3.无状态，在调用一个接口(访问、操作资源)的时候，可以不用考虑上下文，极大的降低了复杂度。
- Application Server:在Java中，为了保证程序能够在各个厂商的AS中兼容运行，Sun公司为制定

了J2EE规范。从技术发展路径来看，Serverlet,JSP演变都是为了更好地方便程序员们编程。以典型的Tomcat为例，Connector和Container组成一个 Service，多个Service组成一个Server。Connector主要负责接受外部请求，Container负责处理请求。Server提供了生命周期管理，如启动，停止等。

- Session Framework：在大型互联网架构中，单台机器已经存放不了用户的登录信息。同时为了支持故障转移等特性，需要一套session管理机制，支持海量用户同时在线。通常可以在遵循J2EE的容器内，使用Filter模式和分布式缓存系统来实现。
- MVC：即Model，View和Controller。Model代表业务逻辑，View表示页面视图，Controller表示根据用户请求，执行相应的业务逻辑，并选择适当的页面视图返回。用过ROR的同学都知道，里面的router配置了什么样的URL和什么样的action相对应。相应的，MVC的本质就是根据不同的URL选择不同的servlet来执行。只不过，结合了Intercepting Filter提供了强大的功能而已。
- IOC：至于为什么需要IOC，笔者在这篇文章进行了讨论。究其本质实现，无非是反射+单例模式+Hash算法+字节码增强+ThreadLocal。前3者用来实现对象生命周期的管理，后2者用来支持AOP，声明式事务。
- ORM：这个词实际上放在这里介绍不太合适，但是笔者目前没想把它单独拉出章节来讲。根据笔者的经验，ORM主要完成了类和表的映射，对象和一条表数据记录的映射。其核心实现是通过jdbc获取数据库的meta信息，然后根据映射关系（这里可以通过COC（Conversion Over Configuration，约定优于配置），注解等技术来简化配置）来动态生成sql和返回数据库的执行结果。

回到顶部

SOA

网站架构的演进之路，从单一应用架构到垂直应用架构，分布式服务架构以及流动计算架构，越来越体现SOA框架的重要性。这里以优秀的开源实现dubbo为例，简单介绍下。

dubbo的功能介绍见服务治理过程，对dubbo架构详细介绍的有如何学习dubbo源代码和dubbo源代码阅读。

简而言之，就是使用了spring的schema的扩展机制，进而支持自定义dubbo标签；通过类似serviceload机制配置多个可选服务。通过jdk动态代理和Javassist，使服务调用透明化。结合ZooKeeper实现高可用元数据管理。

回到顶部

MQ

MQ（Message Queue，消息队列）使服务调用异步化，可以消除并发访问洪峰，提升网站响应速度。在MQ实现中，笔者写过一篇介绍Kafka的学习笔记，详细介绍见Kafka/Metaq设计思想学习笔记，不再多言。

回到顶部

CACHE

Cache就是将数据放到距离计算最近的地方，用来加快处理速度。通常对一定时间内的热点数据进行缓存。
在使用缓存时，需要注意缓存预热和缓存穿透问题。

一般海量数据的缓存系统不会使用Java来实现，是因为Java有额外的对象大小开销以及GC压力。所

以一般是用ANSI C来实现。目前用的比较火的是Redis，更多介绍请查看Redis资料汇总

回到顶部

STORAGE

在出现NOSQL之前，一统天下的是MySQL分库分表技术。结合类似TDDL等SQL agent技术，也能够执行类似join的操作。后来，就像忽如一夜春风来，出现了很多NOSQL/分布式存储系统产品。

分布式存储系统是分布式系统中最复杂的一部分，相比较SOA,CACHE等框架，它需要解决的问题更加复杂。常见的问题如下：

- 数据分布 在多台服务器之间保证数据分布均匀，跨服务器如何读写
- 一致性 异常情况下如何保证副本一致性
- 容错 把发生故障当成常态来设计，做到检测是否发生故障并进行故障迁移
- 负载均衡 新增、移除服务器时如何负载均衡 数据迁移如何不影响已有服务
- 事务并发控制 如何实现分布式事务，如何实现多版本并发控制
- 压缩、解压缩 根据数据特点选择恰当算法，如何平衡时间和空间的关系。

当笔者阅读完《大规模分布式存储系统原理解析与架构实战》和google的两篇存储论文后，感觉里面的实现细节太多了。如果要写的话，还是后面单独列一片把。所以这里暂且略过。

其他

还有其他方面的知识，等后面积累再多些，再重点写吧，这里仅仅是索引下，读者可以自行略过。

- 配置数据、元数据管理系统：可以查看这篇ZooKeeper和Diamond有什么不同
- 搜索系统：机器学习分析用户行为，结合搜索进行推荐排名。各种大数据分析工具。
- 云计算：硬件虚拟化。创业公司可以购买云服务，避免固定资产开销，可能闲置，购买，管理，安装费用，无法迅速购买等问题，属于浮动消费，类似开车和租车的区别，仅是租用服务。云厂商在能源，制冷，运维成本，量大硬件定制，充分利用闲置资源具有优势。
- 鹰眼系统：日志规范化+打点+数据分析+树状展现，详细介绍可以参考 鹰眼下的淘宝-分布式调用跟踪系统介绍
- 系统运维：目标是自动化运维。
  - 监控各种资源指标：
    - OS:(cpu，memory，disk(空间，读写次数))
    - 网络流量
    - 中间件：tomcat，jvm，
    - MQ:通过监控生产者，broker，消费者之间的队列情况，动态决定增加、减少消费者
    - 服务框架自省(运维监控) 依赖关系统计，前台系统访问路径，
  - 显示各种监控结果：Agent —》Explorer，Analyze，Visual，Dashboard，S
  - 预警，运维 自动、手工降级，系统问题自动排查甚至问题自动修复，
- 能源节省：能源消耗(CPU,机柜，水冷)
- 系统安全：涉及系统的方方面面，各种脚本，sql注入，0day等等。
- 版本开发、版本发布：开发环境，测试环境，支持开速发布，不用大的cycle，灰色发布，回滚降级流程，周边协调。大众点评的有个关于开发环境搭建的，感兴趣的可以点击打造高效的单机开发环境。
- 数据中心：在《程序员》2014年第一期介绍里面，提到了阿里使用了ZONE的概念来解决横向扩展的问题。阿里主要是为了解决机房网络瓶颈和超大规模系统的伸缩性问题，把完成某一特

定业务需要的系统、核心服务、数据库组合成一个业务单元。

参考

- 《深入分析Java Web 技术内幕》
- 《大规模分布式存储系统原理解析与架构实战》
- 《大型网站技术架构核心原理与案例分析》
- 《分布式系统原理介绍》
- 《大规模Web服务开发技术》
- 《程序员》2014年第一期
- google系列论文
- varnish / squid / nginx cache 有什么不同？
- RESTful的优点

相关资讯 — 更多

- 互联网技术架构的启示
- 互联网路由架构需要大幅修正
- 信息架构的模式
- 走进支撑过8亿用户的Yahoo!数据中心
- 理解RESTful架构
- 架构师与程序员的区别(图灵访谈)
- 软件架构师
- 可牺牲的架构
- 谈软件架构!!
- 成长型公司如何构建存储架构
- OpenStack架构预览
- 通向架构师之路
- 架构妄想:AJAX + REST
- 专家视角看IT与架构
- Debian移植到OpenRISC架构
- 敏捷和架构的冲突
- 确认:腾讯正式宣布架构重组
- 云计算 OpenNebula的架构
- 业内称12306瘫痪为系统架构规划问题
- 浏览器插件体系架构 FireBreath

相关文档 — 更多

- 大型互联网网站架构心得之一.doc
- 互联网架构分析专场-03金明.pdf
- 互联网架构分析专场-01微博极端压力下的性能优化_唐福林.pdf
- 大型网站技术架构:核心原理与案例分析.pdf
- 大型网站技术架构:核心原理与案例分析.pdf
- 大型网站技术架构 - 核心原理与案例分析.pdf
- 大型网站技术架构:核心原理与案例分析.pdf
- 移动互联网体系架构.doc
- 互联网公司技术架构资料.百度.海量日志分析架构.pdf
- 互联网公司技术架构资料.淘宝.技术架构介绍.pdf
- 互联网公司技术架构资料.腾讯.QQGame后台架构.pdf
- 互联网公司技术架构资料.淘宝.数据库架构演进历程.pdf
- 互联网公司技术架构资料.百度.数据库架构演变与设计.pdf
- 站在电商平台上的互联网金融架构实践.pdf
- 互联网架构分析专场-微博极端压力下的性能优化.pdf
- 互联网公司技术架构资料.腾讯.集中式IT系统的技术架构.pdf
- 又拍网架构中的消息/任务系统(赵钟秋).pdf
- 又拍网架构中的消息,任务系统(赵钟秋).pdf
- 大型网站架构演变.pdf

- 《大型分布式网站架构设计与实践》.pdf

内容信息
4.7
(已有6个人评价)

67%

33%

0%

0%

0%
收藏:8人 发布时间:2014-02-24 22:14:16

经验标签
架构
同类热门经验

- 淘宝的架构
  27613次浏览
- 数据架构规划
  12957次浏览
- 淘宝技术发展(Oracle/支付宝/旺旺)
  6533次浏览
- 淘宝技术发展(分布式时代:服务化)
  5953次浏览
- 采用什么架构，才能够承受大访问量
  7273次浏览
- 浅析MVC模式与三层架构的区别
  12621次浏览

相关经验

- 淘宝的可伸缩高性能互联网架构
  0人评

- 基于Asp.Net平台的开发架构 WebADNuke
  1人评
- 云架构和openstack的思考
  2人评
- 大型网站架构演化
  0人评
- MySQL体系架构概述
  2人评
- 大型网站技术架构的演进
  0人评
- 大型网站系统架构演化之路
  0人评
- 解读大型网站系统架构的演化
  0人评
- 大型网站系统架构的演化
  0人评
- 又拍网架构中的数据库分库设计
  0人评
- 特大型网站技术架构脑图
  0人评
- PHP大型网站的架构实例分析
  12人评
- 第2章 大型网站架构模式
  0人评
- MySQL在大型网站的应用架构演变
  0人评

相关讨论 - 更多

- 大型网站后台架构的Web Server与缓存
- 大型网站架构不得不考虑的10个问题
- 北京知名互联网公司诚聘：互联网高级技术经理（30至45w） ios主
- 北京知名公司招聘：高级、资深java 应用/系统架构师
- 北京多家公司诚聘：java主管 架构师 资深搜索工程师
- 北京招聘多名：资深java、搜索工程师、架构师
- 关于架构的讨论：烦人的细节