

第 1 章

从认识网站开始

本章学习内容：

- 网站概述
- 认识网站类型
- 网站运行原理及开发流程
- 网站的演变过程
- 网站的评估指标
- 什么是集群
- 什么是分布式
- 什么是微服务

1.1 网站概述

Web网站是一个广泛的概念，它可以包括各种类型的网站，如企业官网、资讯类网站、个人博客等。它是一个在线的平台，可以提供各种信息和服务。本书将以Web商城为例进行介绍，Web商城也是Web网站的一种形式，其拥有网站的各种属性，Web商城的特点在于它是一个在线购物平台，类似于淘宝、京东等，它的主要功能是为用户提供商品或服务的购买渠道。所以理解网站的概念和构成对于构建Web商城来说很有必要。

网站是指在互联网上根据一定的规则，使用HTML（标准通用标记语言下的一个应用）等工具制作并用于展示特定内容相关网页的集合。简单地说，网站是一种沟通工具，人们可以通过网站来发布自己想要公开的资讯，或者利用网站来提供相关的网络服务，也可以通过网页浏览器来访问网站，获取自己需要的资讯或者享受网络服务。

在早期，域名（Domain Name）、空间服务器与程序是网站的基本组成部分，随着科技不断进步，网站的组成日趋复杂，目前多数网站由域名、空间服务器、DNS域名解析、网站程序和数据库等组成。

域名由一串用点分隔的字母组成，代表互联网上某一计算机或计算机组的名称，用于在数据传输时标识计算机的电子方位，已经成为互联网的品牌和网上商标保护必备的产品之一。通俗地说，域名就相当于一个家庭的门牌号码，别人通过这个号码很容易找到你所在的位置。以百度的域名为例，百度的网址是由两部分组成的，标号baidu是这个域名的主域名体；前面的www是网络名；最后的标号com则是该域名的后缀，代表这是一个国际域名，属于顶级域名之一。

常见的域名后缀有以下几种。

- .COM: 商业性的机构或公司。
- .NET: 从事Internet相关的网络服务的机构或公司。
- .ORG: 非营利的组织、团体。
- .GOV: 政府部门。
- .CN: 中国国内域名。
- .COM.CN: 中国商业域名。
- .NET.CN: 中国从事Internet相关的网络服务的机构或公司。
- .ORG.CN: 中国非营利的组织、团体。
- .GOV.CN: 中国政府部门。

空间服务器主要有虚拟主机、独立服务器和虚拟专用服务器（Virtual Private Server，VPS）。

虚拟主机是在网络服务器上划分出一定的磁盘空间供用户放置站点和应用组件等，提供必要的站点功能、数据存放和传输功能。所谓虚拟主机，也叫“网站空间”，就是把一台运行在互联网上的服务器划分成多个“虚拟”的服务器。每一个虚拟主机都具有独立的域名和完整的Internet服务器（支持WWW、FTP、E-mail等）。虚拟主机是网络发展的福音，极大地促进了网络技术的应用和普及。同时，虚拟主机的租用服务成了网络时代新的经济形式，虚拟主机的租用类似于房屋租用。

独立服务器是指性能更强大，整体硬件完全独立的服务器，其CPU都在8核以上。

VPS是将一个服务器分区成多个虚拟独立专享服务器的技术。每个使用VPS技术的虚拟独立服务器拥有各自独立的公网IP地址、操作系统、硬盘空间、内存空间和CPU资源等，还可以进行安装程序、重启服务器等操作，与一台独立服务器完全相同。

网站程序是建设与修改网站所使用的编程语言，源代码是由按一定格式书写的文字和符号编写的，可以是任何编程语言，常见的网站开发语言有Java、PHP、ASP.NET、Python和Golang等。而浏览器就如同程序的编译器，它会将源代码翻译成图文内容呈现在网页上。

1.2 认识网站类型

资讯门户类网站以提供信息资讯为主要目的，是目前普遍的网站形式之一，例如新浪、搜狐和新华网。这类网站虽然涵盖的信息类型多、信息量大、访问群体广，但包含的功能比较简单，网站基本功能包含检索、论坛、留言和用户中心等。

这类网站开发主要涉及以下 4 个因素：

- 承载的信息类型，例如是否承载多媒体信息、是否承载结构化信息等。
- 信息发布的方式和流程。
- 信息量的数量级。
- 网站用户管理。

企业品牌类网站用于展示企业综合实力，体现企业文化和品牌理念。企业品牌网站非常强调创意，对于美工设计要求较高，精美的FLASH动画是常用的表现形式。网站内容组织策划和产品展示体验方面也有较高的要求。网站利用多媒体交互和动态网页技术，针对目标客户进行内容建设，以达到品牌营销的目的。

企业品牌网站可细分为以下三类。

- 企业形象网站：塑造企业形象、传播企业文化、推介企业业务、报道企业活动和展示企业实力。
- 品牌形象网站：当企业拥有众多品牌且不同品牌之间的市场定位和营销策略各不相同，企业可根据不同品牌建立其品牌网站，以针对不同的消费群体。
- 产品形象网站：针对某一产品的网站，重点在于产品的体验。

交易类网站以实现交易为目的，以订单为中心。交易的对象可以是企业和消费者。这类网站有3项基本内容：商品如何展示、订单如何生成和订单如何执行。

因此，这类网站一般需要有产品管理、订购管理、订单管理、产品推荐、支付管理、收费管理、送发货管理和会员管理等基本功能。功能复杂一点的可能还需要积分管理系统、VIP管理系统、CRM系统、MIS系统、ERP系统和商品销售分析系统等。交易类网站成功与否的关键在于业务模型的优劣。

交易类网站可细分为以下三大类型。

- B2C（Business To Consumer）网站：商家-消费者，主要是购物网站，用于商家和消费者之间的买卖，如传统的百货商店和购物广场等。
- B2B（Business To Business）网站：商家-商家，主要是商务网站，用于商家之间的买卖，如传统的原材料市场和大型批发市场等。
- C2C（Consumer To Consumer）网站：消费者-消费者，主要以拍卖网站为主，用于个人物品的买卖，如传统的旧货市场、跳蚤市场、废品收购站等。

办公及政府机构网站分为企业办公事务类网站和政府办公类网站。企业办公事务类网站主要包括企业办公事务管理系统、人力资源管理系统和办公成本管理系统。

政府办公类网站是利用政府专用网络和内部办公网络而建立的内部门户信息网，是为了方便办公区域以外的相关部门互通信息、统一处理数据和共享文件资料而建立的，其基本功能有：

- （1）提供多数据源接口，实现业务系统的数据整合。
- （2）统一用户管理，提供方便有效的访问权限和管理权限体系。
- （3）灵活设立子网站，实现复杂的信息发布管理流程。

网站面向社会公众，既可提供办事指南、政策法规和动态信息等，又可提供网上行政业务申报、办理及相关数据查询等。

互动游戏网站是近年来国内逐渐风靡起来的一种网站。这类网站的投入是根据所承载游戏的复杂程度来定的，其发展趋势是向超巨型方向发展，有的已经形成了独立的网络世界。

功能性网站是一种新型网站，其中谷歌和百度是典型代表。这类网站的主要特征是将一个具有广泛需求的功能扩展开来，开发一套强大的功能体系，将功能的实现推向极致。功能在网页上看似简单，但实际投入成本相当惊人，而且效益非常巨大。

1.3 网站运行原理及开发流程

1. 常用术语

如果刚接触网站开发，那么很有必要了解网站的运行原理。在了解网站的运行原理之前，首先需要理解网站中一些常用的术语。

- 客户端：在计算机上运行并连接到互联网的应用程序，简称浏览器，如Chrome、Firefox和IE。用户通过操作客户端实现网站和用户之间的数据交互。
- 服务器：能连接到互联网且具有IP地址的计算机。服务器主要接收和处理用户的请求信息。当用户在客户端操作网页的时候，实质上是向网站发送一个HTTP请求，网站的服务器接收到请求后会执行相应的处理，最后将处理结果返回客户端并生成相应的网页信息。
- IP地址：互联网协议地址，TCP/IP网络设备（计算机、服务器、打印机、路由器等）的数字标识符。互联网上的每台计算机都有一个IP地址，用于识别和通信。IP地址有4组数字，以小数点分隔（例如244.155.65.2），这被称为逻辑地址。为了在网络中定位设备，通过TCP/IP协议将逻辑IP地址转换为物理地址（物理地址即计算机里面的MAC地址）。
- 域名：用于标识一个或多个IP地址。
- DNS：域名系统，用于跟踪计算机的域名及其在互联网上相应的IP地址。
- ISP：互联网服务提供商。主要工作是在DNS（域名系统）中查找当前域名对应的IP地址。
- TCP/IP：传输控制协议/互联网协议，是广泛使用的通信协议。
- HTTP：超文本传输协议，是浏览器和服务器通过互联网进行通信的协议。

2. 网站的运行原理

了解网站常用术语后，我们通过一个简单的例子来讲解网站的运行原理。

- （1）在浏览器中输入网站地址，如www.github.com。
- （2）浏览器解析网站地址中包含的信息，如HTTP协议和域名（github.com）。
- （3）浏览器与ISP通信，在DNS中查找www.github.com所对应的IP地址，然后将IP地址发送到浏览器的DNS服务，最后向www.github.com的IP地址发送请求。
- （4）浏览器从网站地址中获取IP地址和端口（HTTP协议默认为80端口，HTTPS协议默认为443端口），并打开TCP套接字连接，实现浏览器和Web服务器的连接。
- （5）浏览器根据用户操作向服务器发送相应的HTTP请求，如打开www.github.com的主页面。

(6) 当Web服务器接收请求后, 根据请求信息查找该HTML页面。若页面存在, 则Web服务器将处理结果和页面返回浏览器。若服务器找不到页面, 则发送一个404错误消息, 代表找不到相关的页面。

3. 网站的开发流程

很多人认为网站开发是一件很困难的事情, 其实没有想象中那么困难。只要明白了网站的开发流程, 就会觉得网站开发非常简单。如果没有一个清晰的开发流程指导开发, 就会觉得整个开发过程难以实行。完整的开发流程如下。

(1) 需求分析: 当拿到一个项目时, 必须进行需求分析, 清楚知道网站的类型、具体功能、业务逻辑以及网站的风格, 此外还要确定域名、网站空间或者服务器以及网站备案等。

(2) 规划静态内容: 重新确定需求分析, 并根据用户需求规划出网站的内容板块草图。

(3) 设计阶段: 根据网站草图由美工制作成效果图。就好比建房子一样, 首先画出效果图, 然后才开始建房子, 网站开发也是如此。

(4) 程序开发阶段: 根据草图划分页面结构和设计, 前端和后台可以同时进行。前端根据美工效果负责制作静态页面; 后台根据页面结构和设计, 设计数据库的数据结构和开发网站后台。

(5) 测试和上线: 在本地搭建服务器, 测试网站是否存在Bug。若无问题, 则可以将网站打包, 使用FTP上传至网站空间或者服务器。

(6) 维护推广: 在网站上线之后, 根据实际情况完善网站的不足, 定期修复和升级, 以保障网站运营顺畅, 然后对网站进行推广宣传等。

4. 开发任务划分

网站开发必须根据用户需求制定开发任务, 不同职位的开发人员负责不同的功能设计与实现, 各个职位的工作划分如下。

(1) 网页设计由UI负责设计。UI需要考虑用户体验、网站色调搭配和操作流程等。

(2) 前端开发人员将网页设计图转换成HTML页面, 主要编写HTML网页、CSS样式和JavaScript脚本, 如果采用前后端分离, 整个网站的页面功能就皆由前端开发人员实现。

(3) 后端开发人员负责实现网站功能和数据库设计。网站功能需要数据库提供数据支持, 实质上是实现数据库的读写操作; 数据库设计需要根据网站功能设计相应的数据表, 并且还要考虑数据表之间的数据关联。如果采用前后端分离, 后端人员就只需编写API, 由前端人员调用API实现网站功能。

(4) 测试人员负责测试网站功能是否符合用户需求。测试过程需要编写测试用例进行测试, 如果发现功能存在Bug, 就需向开发人员提交Bug的重现方法。只要功能发生修改或变更, 测试人员就要重新测试。

(5) 运维人员负责网站的部署和上线。网站部署主要搭建在Linux系统, 除安装网站所需的运行环境外, 还需要将网站应用搭建在Nginx或Apache服务器上, 并在Nginx或Apache上绑定网站的域名。

1.4 网站的演变过程

随着互联网的不断发展，用户访问和数据量日益增多，使得网站的负载不断增大。如果网站只有一台服务器运行，当负载量达到一定阈值的时候，整个网站可能出现卡顿或崩溃的现象，这时不得不重新设计网站架构。网站的演变应从实际问题出发，从问题中寻找解决方案，实施并调整网站架构。

网站的演变过程可以视为网站架构设计，它可以分为8个阶段，下面分别进行简单介绍。

1. 单机模式

单机模式是指整个网站只部署在一台服务器上，本书的第1~3章都是讲述网站的单机模式的开发和部署。所有网站的演变都是从单机模式开始的，并不是说网站初期不能直接搭建大型架构，只是不太符合实际，毕竟搭建大型网站需要耗费大量的人力和财力。

由于网站初期的用户和数据量较少，并且网站处于0~1阶段，大部分时间主要用于实现网站业务逻辑梳理和功能开发，研发成本主要用于网站功能开发，因此网站架构通常以单机模式为主。

2. Web应用与数据库分离

Web应用与数据库分离是在单机模式的基础上将前端、后端和数据库各自单独部署在一台服务器上。在这种模式下，如果不改变原有的功能，调整网站架构无须修改太多代码，通常修改代码中的一些连接信息，例如Ajax的请求地址、连接数据库的IP地址等即可。

3. 缓存与搜索引擎

缓存与搜索引擎是在数据库出现慢查询的情况下所采用的优化方案之一，慢查询会使网站数据加载出现延时或异常，使得用户体验十分不好。缓存和搜索引擎由后端实现，常见的缓存存储方式有Memcached、数据库、文件和本地内存等；搜索引擎是独立的应用平台，通常支持多种编程语言接入，常用的搜索引擎有Solr和Elasticsearch。

4. 数据库读写分离

数据库读写分离是将数据库的读取和写入分别由两个独立的数据库实现，一个数据库只负责读取数据，另一个数据库只负责写入数据，两个数据库通过数据同步复制保持数据一致。读写分离能提高数据库的负载能力和性能，因为写入比读取需要消耗更多时间，只读取的数据库没有数据写入操作，减轻了磁盘IO等性能问题，所以可以提高数据查询效率。

读写分离需要调整后端代码，至少配置两个或两个以上的数据库连接，同一张数据表的读取和写入分别由不同的数据库连接实现。

5. 数据库拆分

数据库拆分包括水平拆分与垂直拆分。水平拆分是将一个数据表的数据拆分到多个数据表或

多个数据库，也就是我们常说的数据表分表设计；垂直拆分是将一个数据库的所有表拆分到不同数据库，也就是我们常说的数据库分库设计。

6. 集群模式

集群模式是将前端、后端和数据库各自部署到多台服务器，使得一个功能由多台服务器共同完成，以提高网站的负载能力。比如将前端项目部署到服务器A和B，再由同一域名分别解析到服务器A和B，或者通过Nginx或Apache使用负载均衡算法自动分配服务器A和B。

7. 分布式设计

分布式设计是将后端多个功能组件拆分并部署在不同服务器。因为后端功能除使用Web框架外，还会使用其他功能组件，比如搜索引擎、消息队列中间件（Kafka、Redis和RabbitMQ）、文件存储系统等。也就是说，分布式设计是将后端的Web框架和其他功能组件分别部署在不同服务器，彼此之间通过网络连接实现数据通信。后端的各个功能组件除分布式设计外，每个组件还可以实现集群模式。

8. 微服务设计模式

微服务设计模式主要是将后端Web框架实现的功能进一步拆分，拆分后的应用单独部署在服务器中，每个应用之间通过API网关、微服务注册与发现等方式实现调度和通信。以本书的项目为例，后端实现用户登录和产品查询接口，它们都是在同一个后端项目中实现的，如果改为微服务，那么两个接口分别由不同的后端项目实现，接口之间通过API或RPC方式实现通信。

综上所述，我们只是简单介绍了网站每个演变阶段的架构设计方案，每个架构设计方案都是大而全的概念，这是所有网站都适用的设计方案。由于每个网站的功能和业务需求各不相同，因此每个演变阶段的执行方案也各不相同，通俗地讲，同样的食材不同厨师可以烹饪出不同口味的菜式。

1.5 网站评估指标

判断一个网站是否满足当前业务需求可以从5个指标（性能、可用性、伸缩性、扩展性和安全性）来综合评估。

1. 性能

网站的性能是否稳定对于网站的可持续发展有着重要作用。就如人生病一样，网站性能出现问题也可以通过一些指标数据反映出来，评估网站性能的指标有很多，例如CPU占有率、并发量、响应时间、网络传输量、吞吐量、点击率等。

从网站开发的角度来看，网站性能的核心指标有响应时间、并发量和吞吐量，每个指标说明如下：

- 响应时间一般包含网络传输时间和应用程序处理时间，整个过程是从用户发送请求到用户接收服务器返回的响应数据，如果响应时间在3~5s以内，那么表示性能是良好的。

- 并发量是指网站在同一时间的访问人数，进一步细化可以分为业务并发用户数、最大并发访问数、系统用户数、同时在线用户数等。
- 吞吐量是指系统在单位时间内处理请求的数量，其中TPS和QPS都是吞吐量的常用量化指标，服务器的CPU占有率、网络传输速度、外部接口和IO操作等都会影响网站的吞吐量。

除评测网站的各个数据指标外，还有以用户为核心的性能模型RAIL。RAIL分为Response（响应）、Animation（动画）、Idle（浏览器空置状态）和 Load（加载）模块，这是谷歌制定的衡量性能的标准，每个模块的衡量标准如下。

- Response: 网站给用户的响应体验，建议处理事件在50ms内完成。
- Animation: 动画是否流畅，要求每10ms产生一帧。
- Idle: 让浏览器有足够的空闲时间，不能让主线程一直处于繁忙状态。
- Load: 要求5s内完成所有内容的加载并可以交互。

2. 可用性

网站的可用性是指网站出现异常的时候能否正常使用。网站出现异常是一件很正常的事，比如受到黑客攻击、网络故障、DNS劫持、CDN服务异常、程序的Bug等因素，虽然网站异常是无法避免的，但能提高网站的异常处理能力。

提高网站的可用性也称为高可用架构设计，高可用通常采用集群、分布式和微服务注册与发现等技术实现，以确保网站每个应用服务都具备异常处理能力。集群确保某台服务器出现异常的时候，集群内的其他服务器仍能提供正常的服务；分布式确保某个应用服务出现异常之后不会影响到其他应用服务的正常运行；微服务注册与发现确保出现异常的应用服务能被及时发现和处理，以保证网站的每一个用户能够正常访问。

3. 伸缩性

网站的伸缩性是指在突然暴增的负载下能否快速处理，缓解不断上升的用户并发访问压力和不断增长的数据存储需求。以“双十一”为例，各大电商平台的访问量比平常都会多很多，面对这种情况，网站必须有快速处理方案，例如增加集群的服务器数量提高负载能力；“双十一”过后，网站负载降低，应减少集群的服务器数量，节省服务器的费用开支。

提高网站的伸缩性能可以在突发用户需求的情况下，不改变网站原有的架构模式实现快速响应处理，既能保证用户体验，又能降低网站运营成本。网站架构调整都要经历一个研发周期，这涉及到网站代码设计、系统测试和服务器运维等相关工作，在研发周期中，如果网站负载超出负荷，那么只能通过网站伸缩性的解决方案暂缓负荷。

4. 扩展性

网站的扩展性是指网站新增的业务功能对现有功能的影响程度。随着网站的不断发展，功能也会不断扩展，衡量网站的扩展性需要看新增的业务功能是否可以对现有功能透明无影响，不需要修改或者很少改动现有的业务功能就能上线新的业务功能，这要求每个业务功能之间实现低耦合。

提高网站的扩展性最好使用微服务架构，网站的每个应用功能之间互不干扰，彼此之间通过API或RPC方式实现通信。

5. 安全性

网站的安全性确保网站数据不易被窃取，服务器后台不易被黑客入侵，网站不易被攻击。常见的网站攻击有XSS攻击、SQL注入、CSRF攻击、Cookie窃取、DDOS攻击等。

综上所述，网站的性能、可用性、伸缩性、扩展性、安全性都是以用户体验为主的，在互联网世界中，用户就是上帝，若要留住用户，则必须从用户体验和网站功能着手，具体说明如下：

- 用户体验必须确保网站性能良好，网页流畅不卡顿，出现异常也不影响用户使用，保证在用户暴增的情况下也能及时处理负载问题，最后确保用户数据安全，特别是电商平台，这关乎用户资产安全问题。也就是说，保证用户使用流畅已经涉及网站的性能、可用性、伸缩性的架构问题，保证用户数据安全是网站安全性的架构问题。
- 网站功能要不定时更新，毕竟市场永远是动态变化的，为了确保用户不流失或吸纳更多用户，必须根据市场变化调整功能或推出新功能，这涉及网站的扩展性架构问题。

1.6 什么是集群

集群（Cluster）是将一组计算机作为一个总体向用户提供Web应用服务，一组计算机的每个计算机系统是集群的节点。一个理想的集群是用户不知道集群系统的底层节点，在用户看来，集群是一个系统，而非多个计算机系统，而且集群系统的管理员能够任意添加和删改集群系统的节点。

集群并非一个新概念，在20世纪70年代，计算机厂商和研究机构就开始对集群进行研究和开发，主要用于科学计算，所以并未普及开来，直到Linux集群出现，集群概念才得以广为传播。

集群是为了解决单机运算和IO能力不足，提高服务的可靠性和扩展能力，降低整体方案的运维成本。在其他技术不能达到以上目的，或者能达到以上目的，但成本过高的情况下，均可考虑采用集群技术。

按照功能划分，集群分为高可用集群和高性能计算集群。高可用集群简称HA集群，可提供高度可靠的服务；高性能计算集群简称HPC集群，可提供单个计算机不能提供的强大计算能力。

对于网站集群来说，大部分采用高可用集群保证网站的可行性和伸缩性，以确保网站的某个集群节点出现异常仍能提供正常的Web应用服务。高可用集群通常有两种工作方式：容错系统和负载均衡系统，详细说明如下：

- 容错系统通常以主从服务器方式实现，在主服务器正常运行的情况下，从服务器不提供服务，当系统检测发现主服务器出现异常时，从服务器就取代主服务器的工作向外提供服务。
- 负载均衡系统是集群所有节点都正常运行，向外提供服务，它们共同分担整个系统的负载量，这是大型网站常用的技术架构之一。

为什么大型网站通常选择负载均衡系统？

首先考虑成本问题，容错系统在从服务器正常的情况下不提供服务，并且从服务器必须保证在线运行，以确保主服务器出现异常及时切换，服务器运行但不提供服务也要消耗网络、CPU、IO等资源，相当于领着工资不干活，无疑会增加成本开支。

其次考虑负载能力，从服务器不干活，网站所有负载都由主服务器完成，当超出主服务器的负载能力时，用户在使用过程中可能会出现网页白屏、卡顿等情况。

从现实例子理解负载均衡系统，网站相当于一个饭店，顾客相当于用户，网站所有应用功能相当于厨师。假设饭店现有1名厨师并且只能容纳10名顾客，当12名顾客同时光顾饭店时，1名厨师仍能应付。但突然来了20名顾客，为了保证上菜速度，1名厨师就无法烹饪20名顾客的菜肴，只能多聘请1名厨师同时烹饪。只要饭店在营业状态，无论店内有多少名顾客，2名厨师都处于工作状态。

同样的道理，如果网站的一台服务器只能兼容10名用户，当用户数增加到20名时，网站就需要新增一台服务器解决20名用户带来的负载量，并且两台服务器同时提供相同的服务，这就是负载均衡集群。

容错系统相当于饭店聘请了2名厨师，但永远只有厨师A在工作，厨师B就领着工资不干活，当厨师A生病或请假的时候，厨师B才开始工作，只要厨师A上岗，厨师B就不干活。

我们知道网站集群技术主要实现负载均衡，负载均衡在网站的前端、后端和数据库均可实现，其架构如图1-1所示。

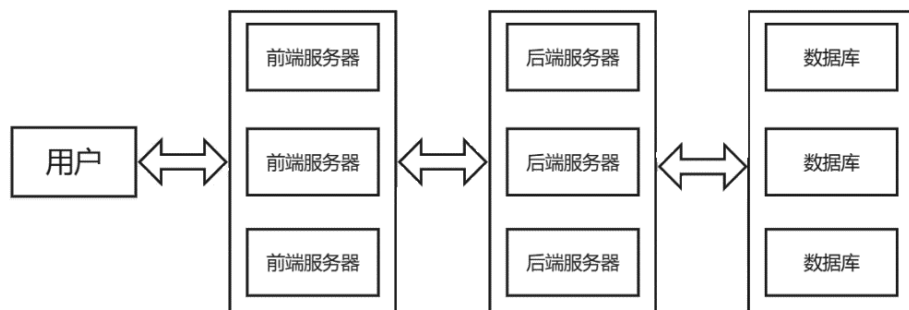


图1-1 网站集群架构

前端的负载均衡可以通过DNS域名解析实现，当我们购买和注册域名的时候，可以将域名绑定到多台前端服务器的IP地址。用户通过浏览器访问域名，DNS服务器解析域名，并通过负载均衡算法将用户请求转发到某台前端服务器的IP地址，再由前端服务器响应用户的HTTP请求。

后端的负载均衡主要通过Nginx或Apache的负载均衡算法实现，当后端收到前端的HTTP请求时，负载均衡服务器将HTTP请求转发到某台Web应用服务器，再由Web应用服务器响应用户的HTTP请求。

数据库的负载均衡主要通过数据库的主从同步方式实现多个数据库之间的数据同步和读写分离。Web应用服务器需要设置多个数据库连接，并执行不同数据表的数据读取和写入，从而实现数据库的负载均衡。

1.7 什么是分布式

网站都是从简到繁逐步发展的，当网站的功能越来越多时，就会使网站的代码目录变得臃肿，代码之间的调用、参数传递等逻辑也会变得复杂，不利于维护和管理，特别是企业的人员流动问题更为突出，新入职的开发人员接手项目也会更为困难。

分布式也称为分布式系统（Distributed System），它将系统的各个功能拆分成多个子系统部署在不同服务器，各个子系统之间通过API或RPC方式实现通信。简单来说，分布式系统就是将系统功能进行拆分，具体包括Web应用和数据库的拆分，拆分方式为水平拆分和垂直拆分。

Web应用的拆分详细说明如下：

- Web应用的水平拆分是将整个应用分层，如将数据库的访问层和业务逻辑层拆分，将网关层和业务逻辑层拆分等。
- Web应用的垂直拆分是按照功能划分子系统，例如将用户管理和产品查询划分为不同的子系统。

数据库的拆分详细说明如下：

- 数据库的水平拆分是将同一个数据表拆分为多个数据表，不同数据表生成在不同数据库中，即我们常说的分库分表。
- 数据库的垂直拆分是按照业务将表进行分类，例如将人员信息表按照性别或居住地拆分为不同数据表存储。

从现实例子理解分布式架构，我们还是以饭店为例，厨师烹饪一道菜肴需要进行清洗食材、准备配料、烹煮食材、菜式摆盘等操作，整个烹饪过程可以看成网站的所有功能，单机模式的网站等于一名厨师独自完成菜肴烹饪；如果是分布式系统，它将菜肴烹饪操作分给不同人员完成，每个工序由不同岗位的员工完成，整个过程如同工厂的生产流水线。

分布式系统通过分而治之的方法使各个子系统独立运行，以提高系统的性能、并发性和可行性。当某个子系统因异常不能使用时，其他子系统还能正常运行；同时，子系统还能采用集群方式提高系统的可用性。

分布式系统主要针对后端和数据库，按照功能划分为不同类别，常见的类别有分布式数据存储、分布式计算、分布式文件系统（Hadoop Distributed File System, HDFS）、分布式消息队列，详细说明如下：

- 分布式数据存储主要针对数据库存储，它主要实现数据库的分库分表，将数据分别存储在不同的数据表或数据库，通过算法将数据尽可能平均地存储到各个数据表或数据库。常用的算法有Hash、一致性Hash、带负载上限的一致性Hash、带虚拟节点的一致性Hash和分片。除此之外，分布式数据存储还要考虑分布式ID生成和分布式事务处理等数据处理问题。
- 分布式计算把计算任务拆分成多个小的计算任务，并且分布在不同的服务器计算，再进行结果汇总，例如淘宝“双十一”实时计算各地区的消费情况。分布式计算通常采用分片算法、消息队列和Hadoop的MapReduce实现，其核心在于计算任务的拆分思维。
- 分布式文件系统将数据文件分散到不同节点或服务器存储，大大减小了数据丢失的风险。常见的分布式文件系统有FastDFS、GFS、HDFS、Ceph、GridFS、MogileFS、TFS等。
- 分布式消息队列主要解决Web应用耦合、异步消息、流量削峰等问题，可实现高性能、高可用、可伸缩的网站架构。消息队列中间件是分布式消息队列的重要组件，常用的消息队列中间件有ActiveMQ、RabbitMQ、ZeroMQ、Kafka、MetaMQ、RocketMQ。

从字面上简单理解，分布式系统是将网站功能拆分为各个子系统独立运行，但深入了解之后，才会发现背后庞大的技术体系，整个技术体系只为保证各个子系统之间协调工作。

1.8 什么是微服务

微服务（Microservice）是一种架构概念，它将功能分解成不同的服务，以降低系统的耦合性，提供更加灵活的服务支持，各个服务之间通过API进行通信。从微服务架构的设计模式来看，它包含开发、测试、部署和运维等多方面的因素。

从概念来看，分布式和微服务十分相似，微服务也是拆分网站功能，但它只对系统执行垂直拆分，并且拆分粒度更细，每个服务自成一体，具有较强的兼容性。举个例子，以用户管理为例，分布式的用户管理只适用于系统A，但微服务的用户管理不仅适用于系统A，还适用于其他系统。

对于大型网站来说，微服务架构可以将网站功能拆分为多个不同的服务，每个服务部署在不同的服务器上，每个服务之间通过API实现数据通信，从而构建网站功能。

服务之间的通信需要考虑服务的部署方式，比如重试机制、限流机制、熔断机制、负载均衡机制和缓存机制等因素，这样能保证每个服务之间的稳健性。

微服务架构有6种常见的设计模式，每种模式的设计说明如下。

1. 聚合器微服务设计模式

聚合器调用多个微服务实现应用程序或网页所需的功能，每个微服务都有自己的缓存和数据库，这是一种常见的、简单的设计模式，其设计原理如图1-2所示。

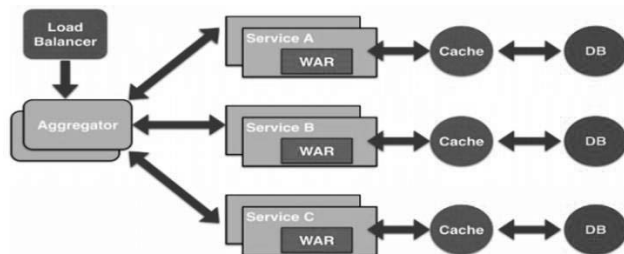


图1-2 聚合器微服务设计模式

2. 代理微服务设计模式

这是聚合器微服务设计模式的演变模式，应用程序或网页根据业务需求的差异而调用不同的微服务，代理可以委派HTTP请求，也可以进行数据转换工作，其设计原理如图1-3所示。

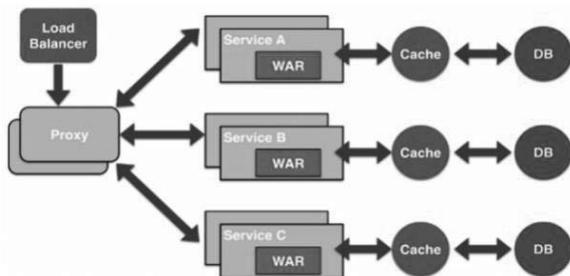


图1-3 代理微服务设计模式

3. 链式微服务设计模式

每个微服务之间通过链式方式进行调用，比如微服务A接收到请求后会与微服务B进行通信，类似地，微服务B会与微服务C进行通信，所有微服务都使用同步消息传递。在整个链式调用完成之前，浏览器会一直处于等待状态，其设计原理如图1-4所示。

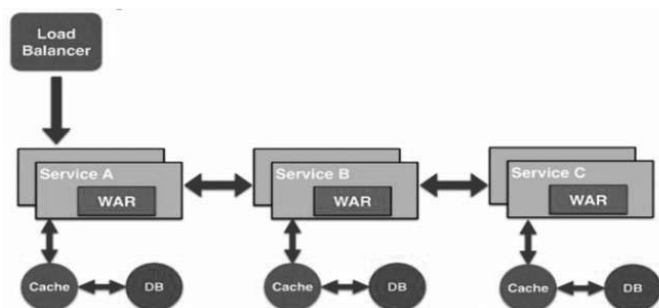


图1-4 链式微服务设计模式

4. 分支微服务设计模式

这是聚合器微服务设计模式的扩展模式，允许微服务之间相互调用，其设计原理如图1-5所示。

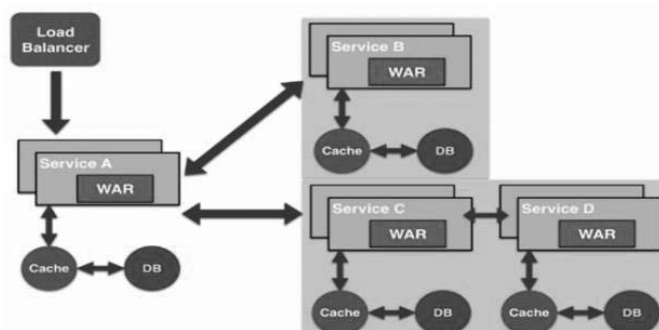


图1-5 分支微服务设计模式

5. 数据共享微服务设计模式

部分微服务可能会共享缓存和数据库，即两个或两个以上的微服务共用一个缓存和数据库。这种情况只有在两个微服务之间存在强耦合关系时才能使用，对于使用微服务实现的应用程序或网页而言，这是一种反模式设计，其设计原理如图1-6所示。

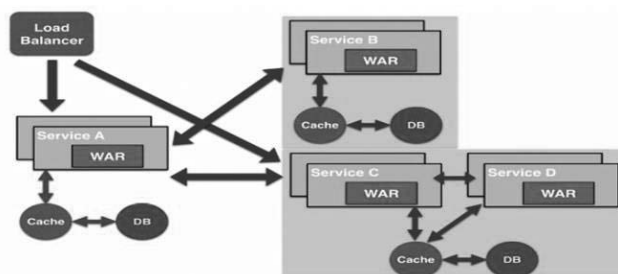


图1-6 数据共享微服务设计模式

6. 异步消息传递微服务设计模式

由于API使用同步模式，如果API执行的程序耗时过长，就会增加用户的等待时间，因此某些微服务可以选择使用消息队列（异步请求）代替API的请求和响应，其设计原理如图1-7所示。

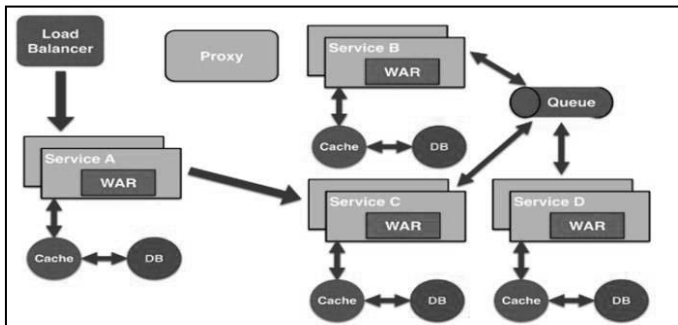


图1-7 异步消息传递微服务设计模式

微服务设计模式不是唯一的，具体还需要根据项目需求、功能和应用场景等多方面因素综合考虑。对于微服务架构，架构的设计意识比技术开发更为重要，整个架构设计需要考虑多个微服务的运维难度、系统部署依赖、微服务之间的通信成本、数据一致性、系统集成测试和性能监控等。

1.9 本章小结

网站是指在互联网上根据一定的规则，使用HTML（标准通用标记语言下的一个应用）等工具制作并用于展示特定内容相关网页的集合。在早期，域名、空间服务器与程序是网站的基本组成部分，随着科技的不断进步，网站的组成也日趋复杂，目前多数网站由域名、空间服务器、DNS域名解析、网站程序和数据库等组成。

网站开发流程如下。

- (1) 需求分析: 当拿到一个项目时, 必须进行需求分析, 清楚知道网站的类型、具体功能、业务逻辑以及网站的风格, 此外还要确定域名、网站空间或者服务器以及网站备案等。
- (2) 规划静态内容: 重新确定需求分析, 并根据用户需求规划出网站的内容板块草图。
- (3) 设计阶段: 根据网站草图, 由美工制作成效果图。就好比建房子一样, 首先画出效果图, 然后才开始建房子, 网站开发也是如此。
- (4) 程序开发阶段: 根据草图划分页面结构和设计, 前端和后台可以同时进行。前端根据美工效果负责制作静态页面; 后台根据页面结构和设计, 设计数据库的数据结构和开发网站后台。
- (5) 测试和上线: 在本地搭建服务器, 测试网站是否存在Bug。若无问题, 则可以将网站打包, 使用FTP上传至网站空间或者服务器。
- (6) 维护推广: 在网站上线之后, 根据实际情况完善网站的不足, 定期修复和升级, 保障网站运营顺畅, 然后对网站进行推广宣传等。

网站的演变过程可以视为网站架构设计，它可以分为8个阶段：

- (1) 单机模式。
- (2) Web应用与数据库分离。
- (3) 缓存与搜索引擎。
- (4) 数据库读写分离。
- (5) 数据库拆分。
- (6) 集群模式。
- (7) 分布式设计。
- (8) 微服务设计模式。

判断网站是否满足当前业务需求可以从5个指标（性能、可用性、伸缩性、扩展性和安全性）综合评估，从网站开发角度来看，网站性能的核心指标主要有响应时间、并发量和吞吐量。

高可用集群通常有两种工作方式：容错系统和负载均衡系统。

- 容错系统通常以主从服务器方式实现，在主服务器正常运行的情况下，从服务器不提供服务，当系统检测发现主服务器出现异常时，从服务器就取代主服务器的工作向外提供服务。
- 负载均衡系统是集群所有节点都正常运行，向外提供服务，它们共同分担整个系统的负载量，这是大型网站常用的技术架构之一。

分布式系统就是将系统功能进行拆分，具体包括Web应用和数据库拆分，拆分方式为水平拆分和垂直拆分。

Web应用的拆分详细说明如下：

- Web应用的水平拆分是将整个应用分层，如将数据库访问层和业务逻辑层拆分，将网关层和业务逻辑层拆分等。
- Web应用的垂直拆分是按照功能划分子系统，例如将用户管理和产品查询划分为不同的子系统。

数据库的拆分详细说明如下：

- 数据库的水平拆分是将同一个数据表拆分为多个数据表，不同数据表生成在不同数据库中，即我们常说的分库分表。
- 数据库的垂直拆分是按照业务将表进行分类，例如将人员信息表按照性别或居住地拆分为不同数据表存储。

微服务是一种架构概念，它将功能分解成不同的服务，以降低系统的耦合性，提供更加灵活的服务支持，各个服务之间通过API进行通信。从微服务架构的设计模式来看，它包含开发、测试、部署和运维等多方面因素。

从概念来看，分布式和微服务十分相似，微服务也是拆分网站功能，但它只对系统执行垂直拆分，并且拆分粒度更细，每个服务自成一体，具有较强的兼容性。