**网站服务器配置如何选择**

　　1. 初始阶段的网站架构

　　一般来讲，大型网站都是从小型网站发展而来，一开始的架构都比较简单，随着业务复杂和用户量的激增，才开始做很多架构上的改进。当它还是小型网站的时候，没有太多访客，一般来讲只需要一台服务器就够了，这时应用程序、数据库、文件等所有资源都在一台服务器上，网站架构如下图所示：

　　常见的网站服务器架构有哪些？  
　　2. 应用服务和数据服务分离

　　随着网站业务的发展和用户量的增加，一台服务器就无法再满足需求了。大量用户访问导致访问速度越来越慢，而逐渐增加的数据也会导致存储空间不足。这时就需要将应用和数据分离，应用和数据分离后整个网站使用 3 台服务器：应用服务器、文件服务器和数据库服务器。这 3 台服务器对硬件资源的要求各不相同：

　　应用服务器业务逻辑，需要强大的CPU  
　　数据库服务器对磁盘读写操作很多，需要更快的磁盘和更大的内存  
　　文件服务器存储用户上传的文件，因此需要更大的磁盘空间  
　　此时，网站系统的架构如下图所示：

　　常见的网站服务器架构有哪些？  
　　3. 使用缓存改善网站性能

　　随着用户再增加，网站又会一次面临挑战：数据库压力太大导致整站访问效率再此下降，用户体验受到影响。一个网站，往往 80% 的业务访问集中在 20% 的数据上，比如微博请求量最多的肯定是那些千万级粉丝的大 V 的微博，而几乎没有人关注的你的首页，除了自己想起来之外根本不会被打开。既然大部分业务访问集中在一小部分数据上，那就把这一小部分数据先提前缓存在内存中，而不是每次都去数据库读取，这样就可以减少数据库的访问压力，从而提高整个网站的访问速度。

　　网站使用的缓存一般分为缓存到应用服务器或者缓存在专门的分布式缓存服务器。缓存到应用服务器自己的访问速度快很多，但是受自身内存限制，往往不太适用。远程分布式缓存使用一个集群专门负责缓存服务，当内存不够还可以轻松得动态扩容。

　　常见的网站服务器架构有哪些？  
　　4. 使用应用服务器集群改善网站的并发处理能力

　　使用缓存后，数据访问压力得到了缓解，但是单一应用服务器能够处理的请求连接有限，在网站访问高峰期，应用服务器就成了整个网站的效率瓶颈。使用分布式集群是网站解决高并发、海量数据问题的常用手段。当一台服务器的处理能力和存储空间不足时，不要尝试去更换更强大的服务器，对大型网站而言，多么强大的服务器，都满足不了网站持续增长的业务需求。这种情况下，更恰当的做法是增加一台服务器分担原有服务器的访问及存储压力。 对网站架构而言，只要能通过增加一台服务器的方式改善负载压力，就可以以同样的方式持续增加服务器不断改善系统性能，从而实现系统的可伸缩性。应用服务器实现集群是网站可伸缩架构设计中较为简单成熟的一种，如下图所示：

　　常见的网站服务器架构有哪些？  
　　通过负载均衡调度服务器，可以将来自用户浏览器的访问请求分发到应用服务器集群中的任何一台服务器上，如果有更多用户，就在集群中加入更多的应用服务器，使应用服务器的压力不再成为整个网站的瓶颈。

　　5. 数据库读写分离

　　网站在使用缓存后，使对大部分数据读操作访问都可以不通过数据库就能完成，但是仍有一部分读操作（缓存访问不命中、缓存过期）和全部的写操作都需要访问数据库，在网站的用户达到一定规模后，数据库因为负载压力过高而成为网站的瓶颈。 目前大部分的主流数据库都提供主从热备功能，通过配置两台数据库主从关系，可以将一台数据库服务器的数据更新同步到另一台服务器上。网站利用数据库的这一功能，实现数据库读写分离，从而改善数据库负载压力。如下图所示：

　　应用服务器在写数据的时候，访问主数据库，主数据库通过主从复制机制将数据更新同步到从数据库，这样当应用服务器读数据的时候，就可以通过从数据库获得数据。为了便于应用程序访问读写分离后的数据库，通常在应用服务器端使用专门的数据访问模块，使数据库读写分离对应用透明。

　　常见的网站服务器架构有哪些？  
　　6. 使用反向代理和 CDN 加速网站响应

　　随着网站业务不断发展，用户规模越来越大，由于中国复杂的网络环境，不同地区的用户访问网站时，速度差别也极大。有研究表明，网站访问延迟和用户流失率正相关，网站访问越慢，用户越容易失去耐心而离开。为了提供更好的用户体验，留住用户，网站需要加速网站访问速度。主要手段有使用 CDN 和反向代理。如下图所示：

　　常见的网站服务器架构有哪些？  
　　7. 使用分布式文件系统和分布式数据库系统

　　任何强大的单一服务器都满足不了大型网站持续增长的业务需求。数据库经过读写分离后，从一台服务器拆分成两台服务器，但是随着网站业务的发展依然不能满足需求，这时需要使用分布式数据库。文件系统也一样，需要使用分布式文件系统。如下图所示：

　　常见的网站服务器架构有哪些？  
　　分布式数据库是网站数据库拆分的最后手段，只有在单表数据规模非常庞大的时候才使用。不到不得已时，网站更常用的数据库拆分手段是业务分库，将不同业务的数据部署在不同的物理服务器上。

　　8. 使用 NoSQL 和搜索引擎

　　随着网站业务越来越复杂，对数据存储和检索的需求也越来越复杂，网站需要采用一些非关系数据库技术如 NoSQL 和非数据库查询技术如搜索引擎。如下图所示：

　　常见的网站服务器架构有哪些？  
　　NoSQL 和搜索引擎都是源自互联网的技术手段，对可伸缩的分布式特性具有更好的支持。应用服务器则通过一个统一数据访问模块访问各种数据，减轻应用程序管理诸多数据源的麻烦。

　　9. 业务拆分

　　大型网站为了应对日益复杂的业务场景，通过使用分而治之的手段将整个网站业务分成不同的产品线。如大型购物交易网站都会将首页、商铺、订单、买家、卖家等拆分成不同的产品线，分归不同的业务团队负责。

　　具体到技术上，也会根据产品线划分，将一个网站拆分成许多不同的应用，每个应用独立部署。应用之间可以通过一个超链接建立关系（在首页上的导航链接每个都指向不同的应用地址），也可以通过消息队列进行数据分发，当然最多的还是通过访问同一个数据存储系统来构成一个关联的完整系统，如下图所示：

　　常见的网站服务器架构有哪些？  
　　10. 分布式服务

　　随着业务拆分越来越小，存储系统越来越庞大，应用系统的整体复杂度呈指数级增加，部署维护越来越困难。由于所有应用要和所有数据库系统连接，在数万台服务器规模的网站中，这些连接的数目是服务器规模的平方，导致数据库连接资源不足，拒绝服务。

　　既然每一个应用系统都需要执行许多相同的业务操作，比如用户管理、商品管理等，那么可以将这些共用的业务提取出来，独立部署。由这些可复用的业务连接数据库，提供共用业务服务，而应用系统只需要管理用户界面，通过分布式服务调用共用业务服务完成具体业务操作。如下图所示：

　　常见的网站服务器架构有哪些？  
　　大型网站的架构演化到这里，基本上大多数的技术问题都可以得以解决了。

　　1 SpringBoot+ 高并发消息处理 EDM?项目 实战

　　2 SpringBoot ELK?分布式 数据分析

　　3 Netty?高 并发 UTS?项目实战

　　4 SpringCloud?微服务+NoSQL+ 负载均衡平台设计