

一、概述

1. 大型网站软件系统的特点

- 高并发、大流量
- 高可用
- 海量数据
- 用户分布广泛，网络情况复杂
- 安全环境恶劣
- 需求快速变更，发布频繁
- 渐进式发展

2. 大型网站架构演化发展历程

- 初始阶段

应用程序，文件和数据库部署在一台服务器上。

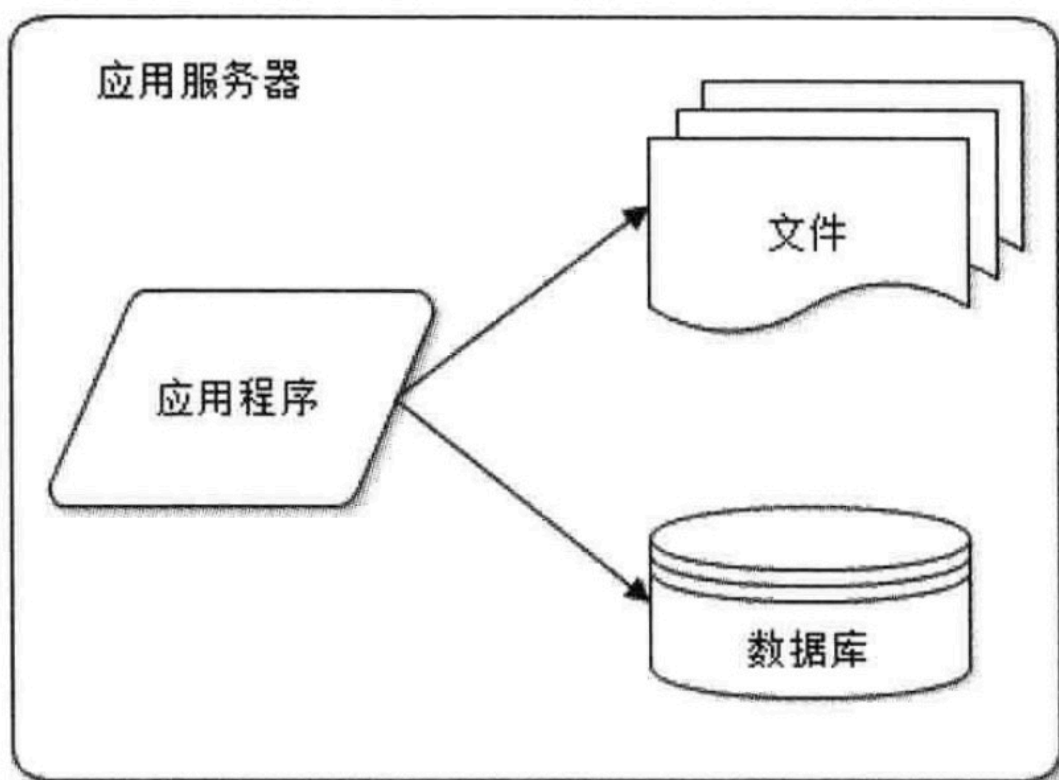


图 1.1 初始阶段的网站架构

- 应用服务和数据服务分离

随着网站的发展，网站性能越来越差，数据量越来越大。需要将原有的单台服务器扩展至多台。

- 应用服务器，处理业务逻辑，需要强大的CPU

- 数据服务器，检索硬盘和缓存数据，需要更快的硬盘和更大的内存
- 文件服务器，存储用户上传的文件数据，需要更大的硬盘

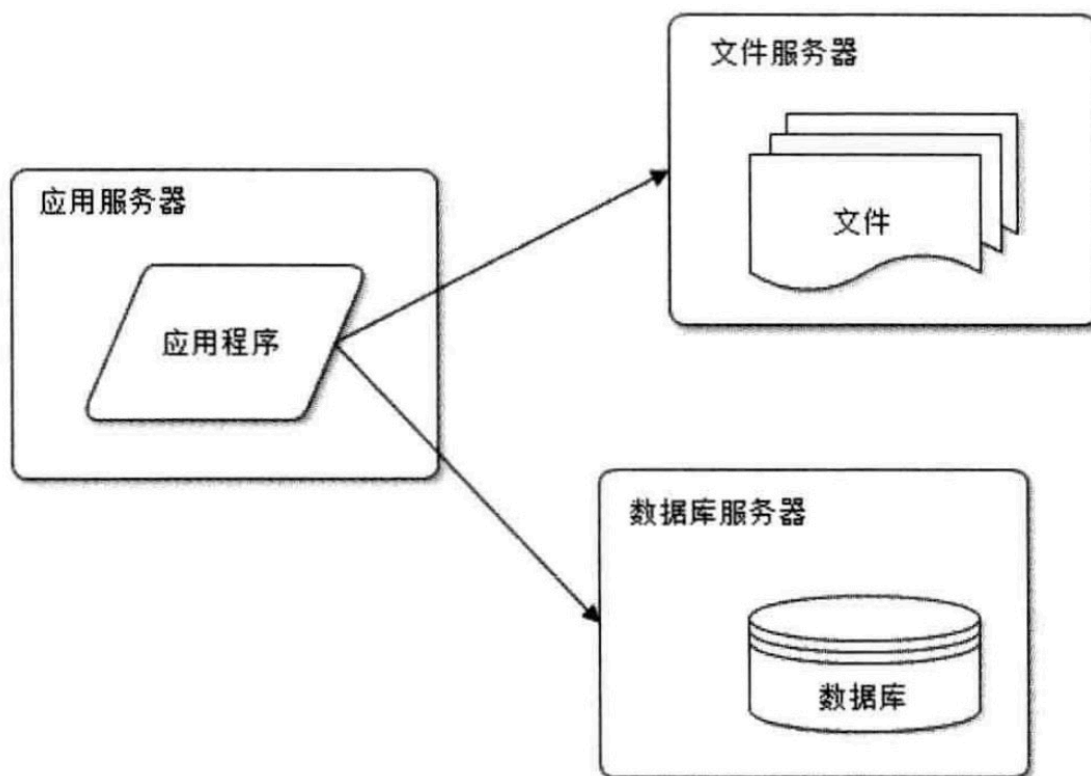


图 1.2 应用服务和数据服务分离

- 使用缓存改善网站性能

二八原则：80%的业务访问在20%的数据上。为了减少对数据库访问的压力，提高网站的访问速度，需要对热点数据做缓存。

- 本地缓存：将数据缓存到应用服务器上，访问更快，但是受到应用服务器内存限制。
- 远程分布式缓存：可以使用集群的方式，部署大内存的服务器作为专门的缓存服务器，理论上做到不受内存容量限制的缓存服务。

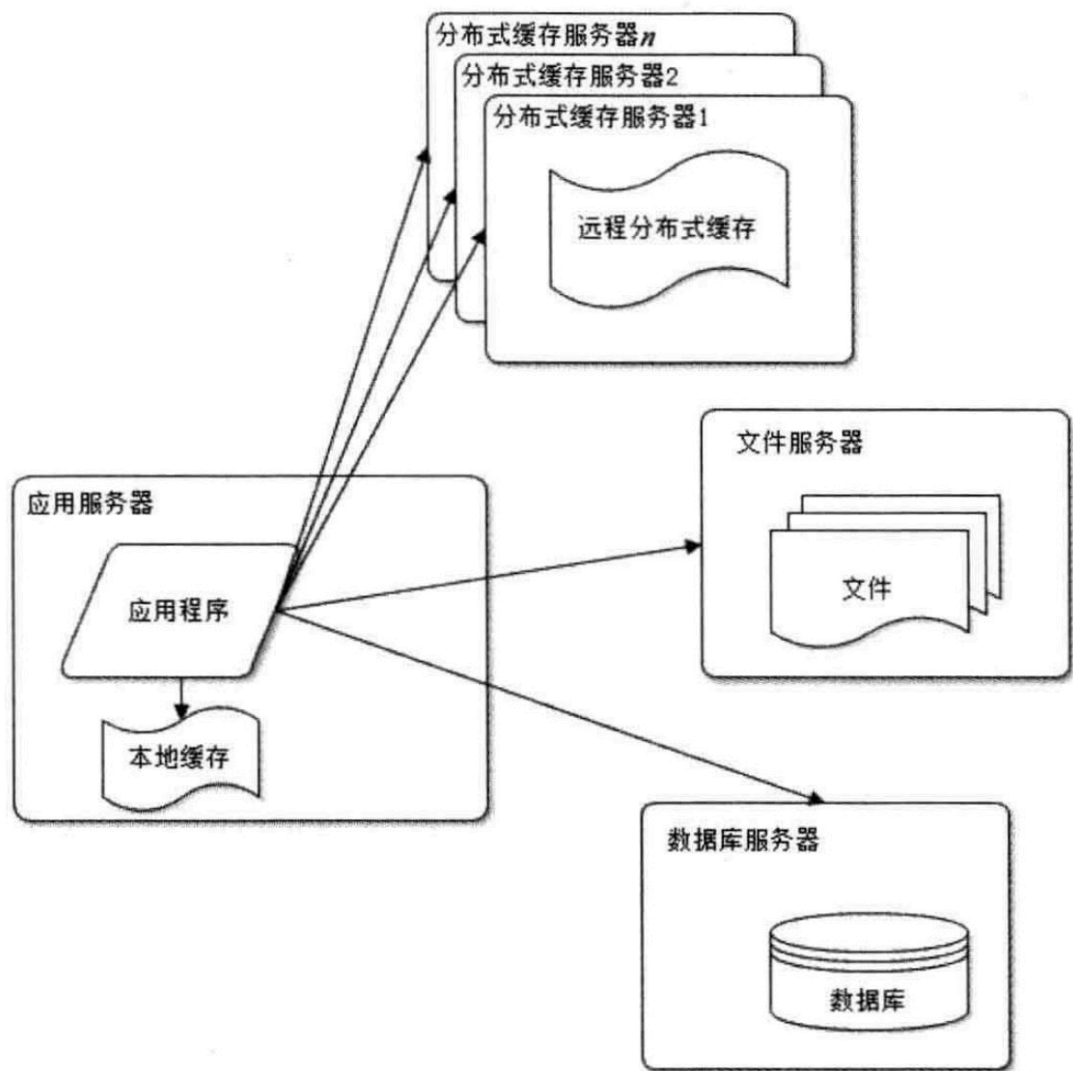


图 1.3 网站使用缓存

- 使用应用服务器集群的方式改善网站并发处理能力

当一台服务器的处理能力和存储能力不足以满足网站的业务需求时，可以通过增加服务器做业务集群，再使用负载均衡将用户需求均摊到各服务器上，从而实现系统的伸缩性。

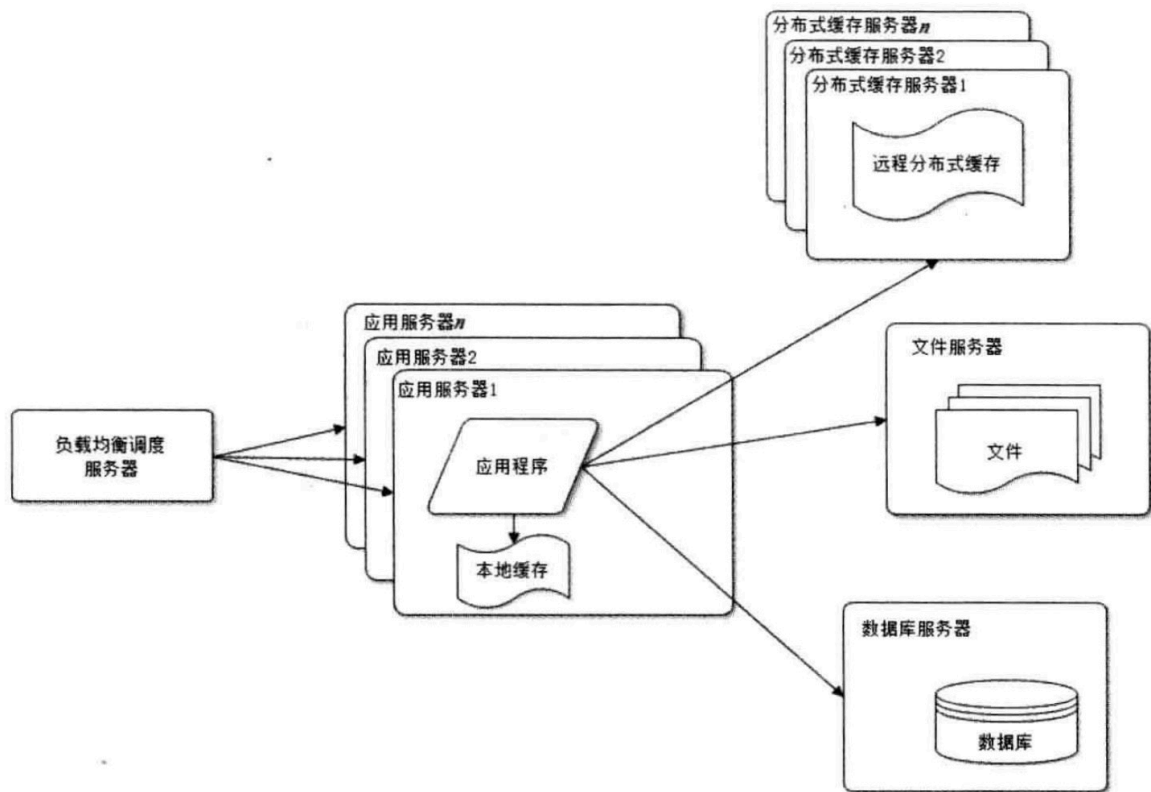


图 1.4 应用服务器集群部署

- 数据库读写分离

对于数据库而言，承担全部的写操作，和部分的读操作（缓存不命中，缓存失效等）。当网站达到一定的规模，数据库因为负载过高成为网站的瓶颈。

大多数主流数据库提供主从复制的功能，可以设置一台主库，当实时复制数据到从库。主库承担写操作，从库承担读操作。

为了便于数据库访问读写分离后的数据库，通常应用层使用专门的数据访问模块，使数据库的读写分离对应用透明。

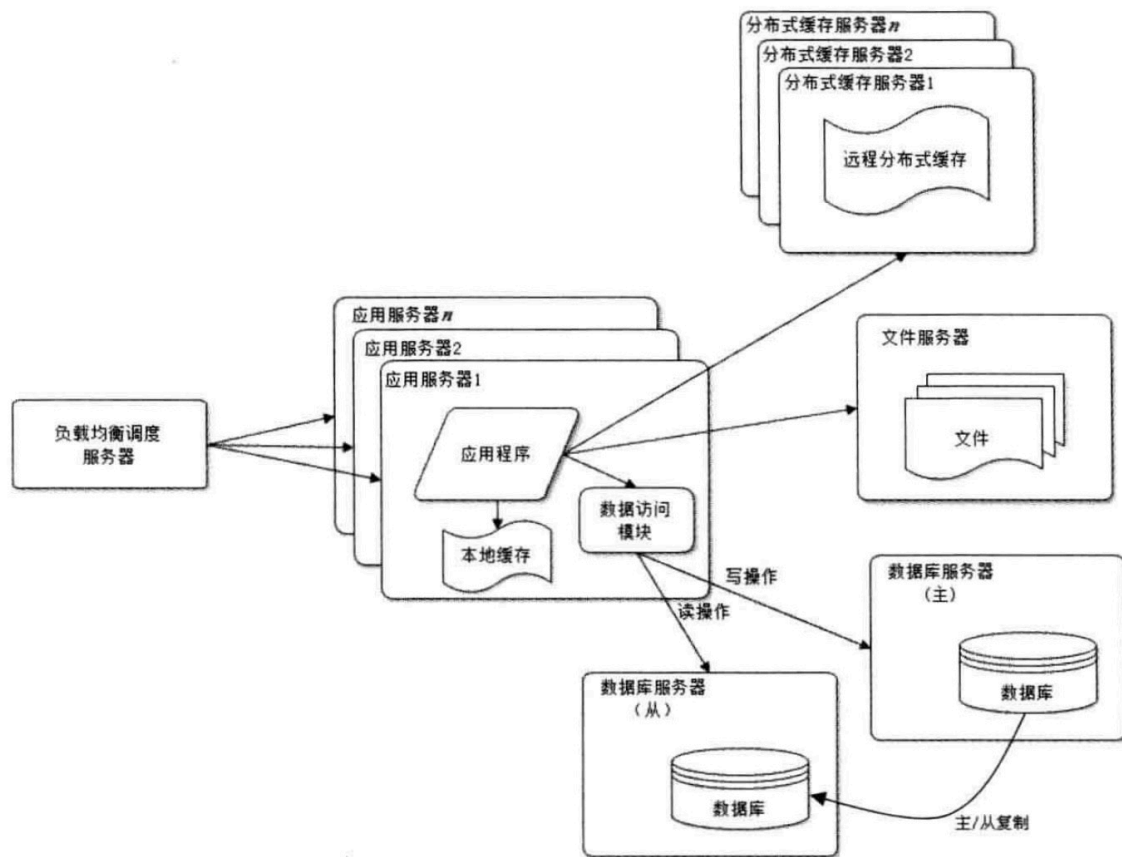


图 1.5 数据库读写分离

- 使用反向代理和CDN加速网站响应

反向代理和CDN的原理都是基于缓存，皆能提高网站响应速度，并减轻服务端压力。

- 反向代理：部署在网站的中心机房。当用户请求到达中心机房，首先访问反向代理服务器，如果该服务器缓存了用户请求需要的数据，直接返回。
- CDN：部署在网络提供商的机房。用户分布范围较大，在请求数据时，可以从距离自己最近的机房拿到数据。

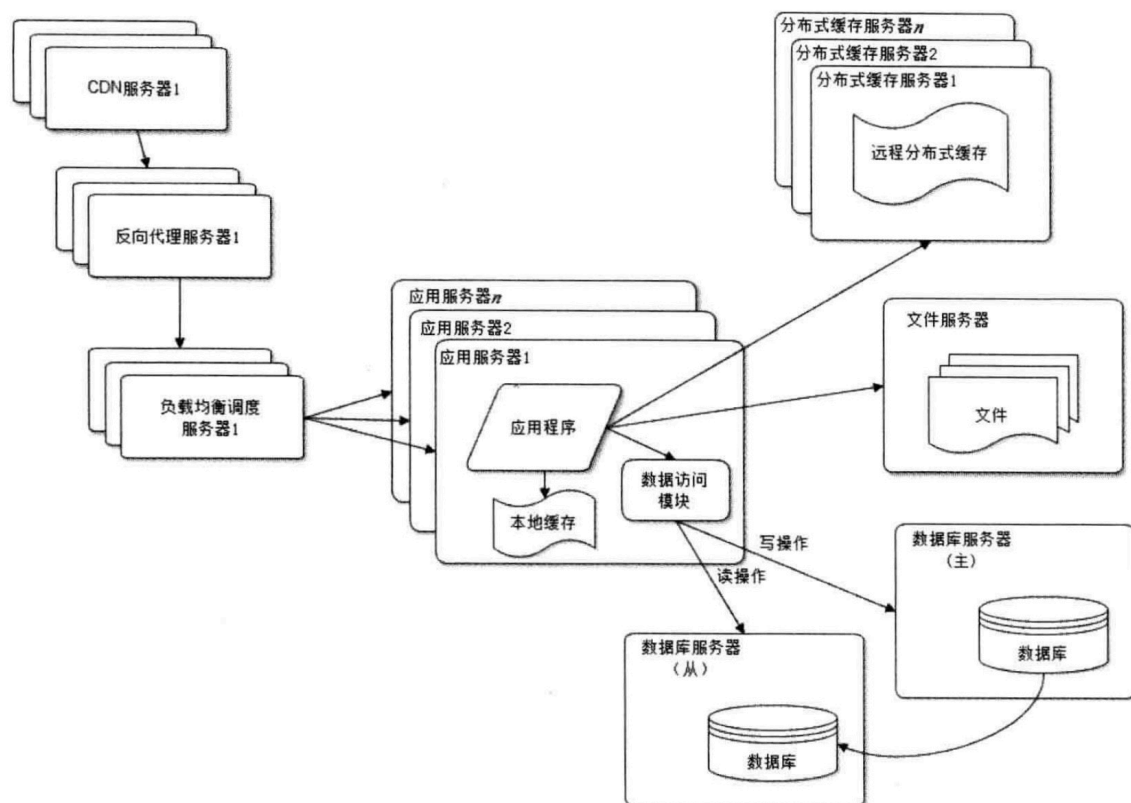


图 1.6 网站使用反向代理和 CDN 加速访问

- 使用分布式文件系统和分布式数据库系统

数据库经过主从分离后，性能依然会是网站最大的瓶颈，这时需要分布式数据库系统，分布式文件系统同理。

分布式数据库系统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，更常见的措施是拆分业务数据库，将不同业务使用的数据分别存储到不同的机器上。

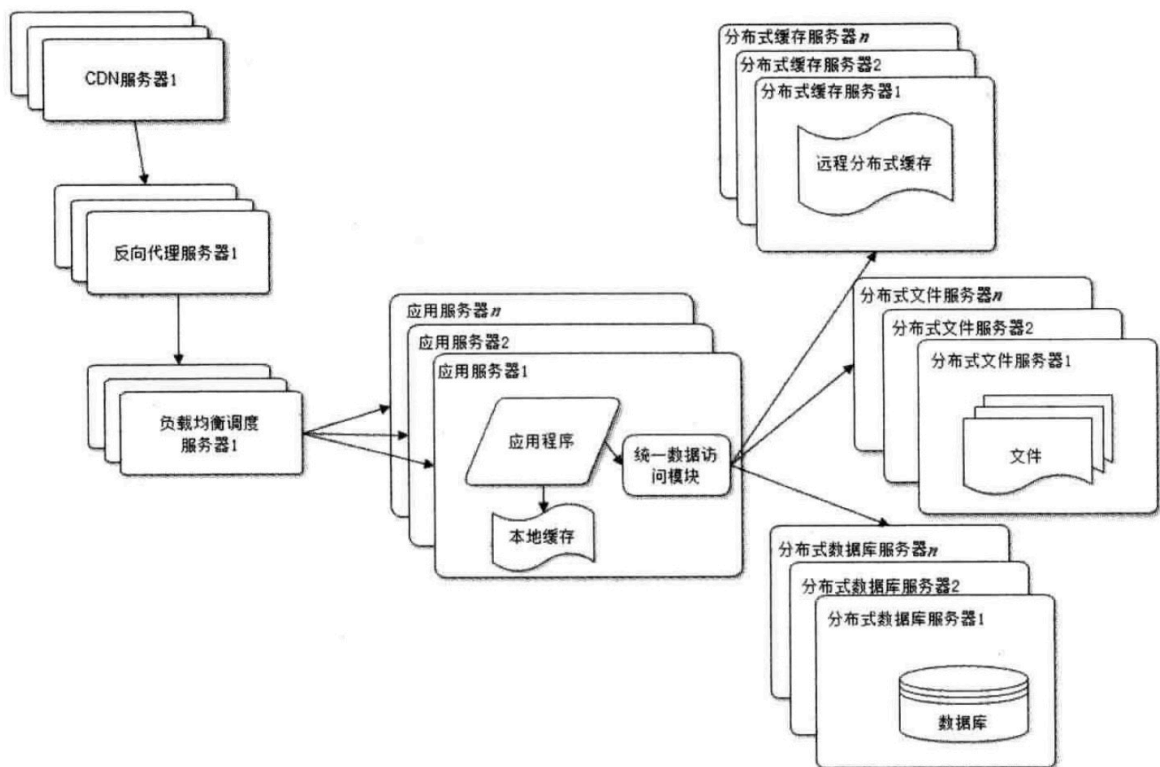


图 1.7 使用分布式文件和分布式数据库系统