# 第六章系统优化项之页面技术

缓存预热的实现

同步抢购倒计时

使用 CDN 技术进行优化

使用页面静态化技术减轻服务器负担

## 1. 缓存预热的实现

为什么需要缓存预热？

版权归属于九章算法（杭州）科技有限公司，贩卖和传播盗版将被追究刑事责任 第4页

### 1.1 缓存预热实现步骤分析

#### 

01

将秒杀活动信息，商品信息，写入redis缓存中

访问秒杀商品详情页面时，

02

先去缓存查询是否有缓存的信息，如果没有再去数

据库中查询

### 1.2 缓存预热实现

#### 1.2.1 秒杀商品详情改造

缓存预热代码实现



#### 1.2.2 添加接口, 将指定的商品信息和活动写入Redis

缓存预热代码实现

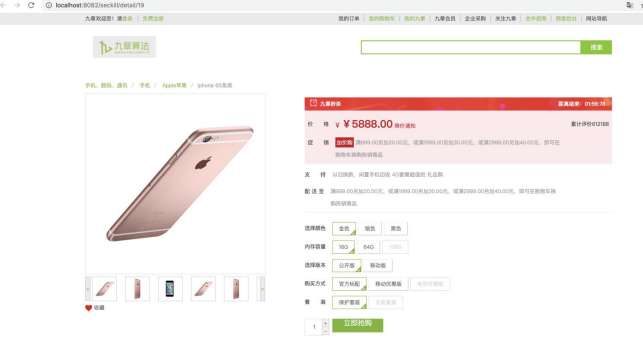
#### 1.2.3 执行测试

缓存预热代码实现



#### 1.2.4 测试页面访问

访问秒杀商品详情页面时，先去缓存查询了缓存中的数据



#### 1.2.5 控制台打印查询缓存结果

查看是否命中缓存

## 2. 同步抢购倒计时

### 2.1 方案对比

如何计算倒计时？

为了防止刷爆商品页面，未开始抢购时，禁用抢购按钮

方法 1 ： 打开页面获取活动开始时间，然后前端页面开始倒计时

方法 2 ：打开页面获取距离活动开始的时间差，然后前端页面，开始倒计时

方法 3 ：前端轮询(Poll)服务器的时间，并获取距离活动开始的时间差



### 2.2 代码开发

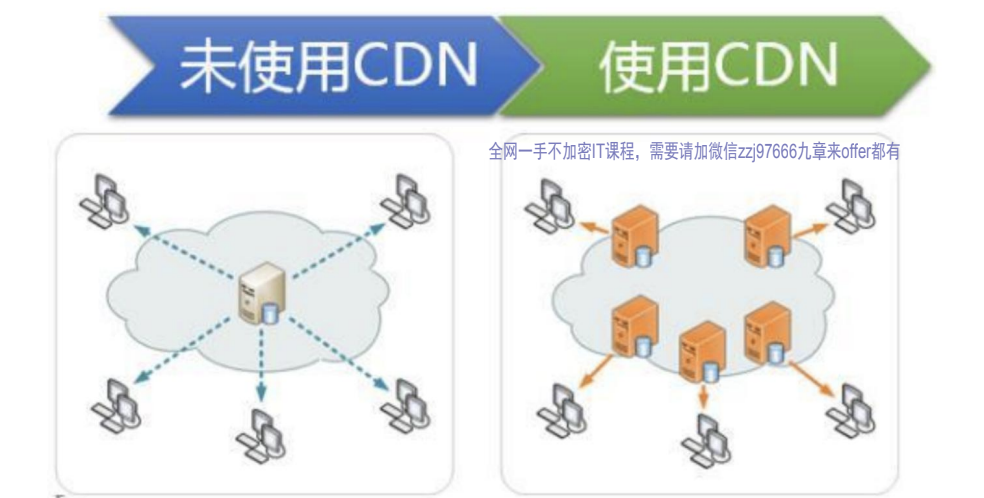
后端添加一个接口向前端提供时间

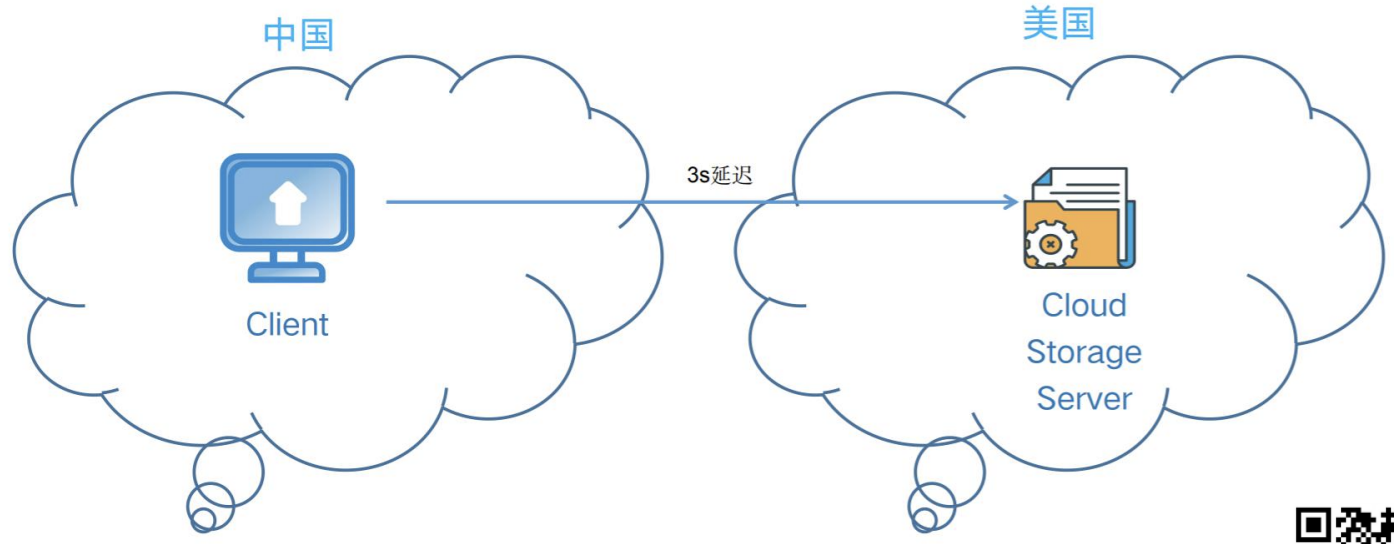
## 3. 使用CDN技术进行优化CDN（Content Delivery Network）

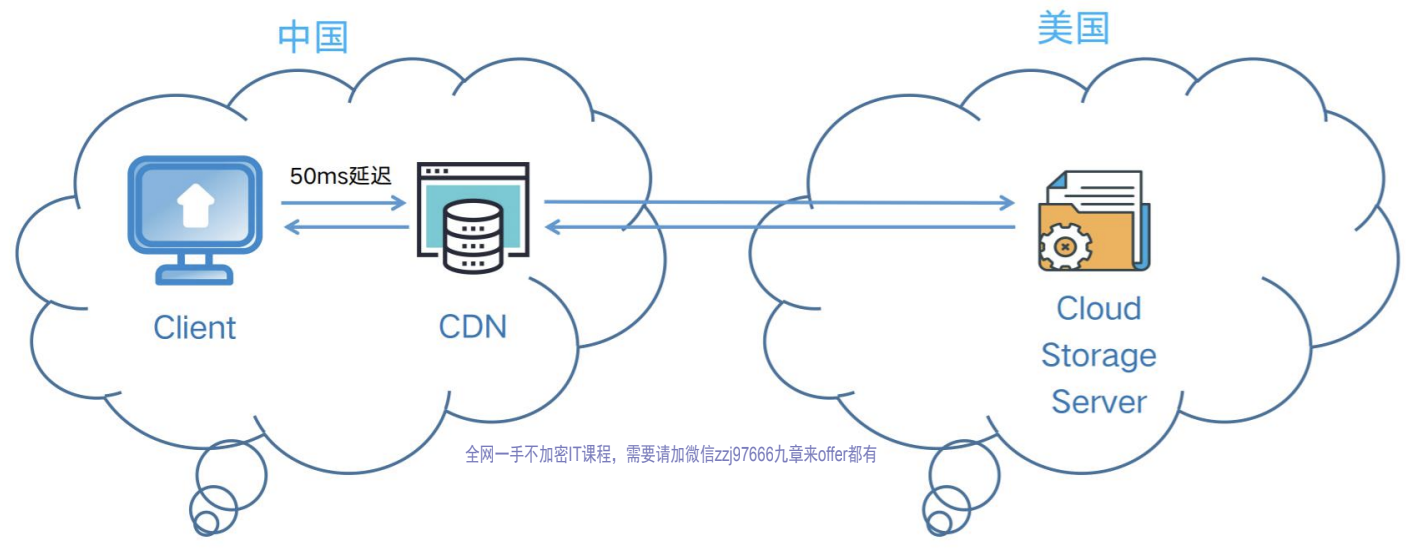
### 为什么要使用CDN技术？

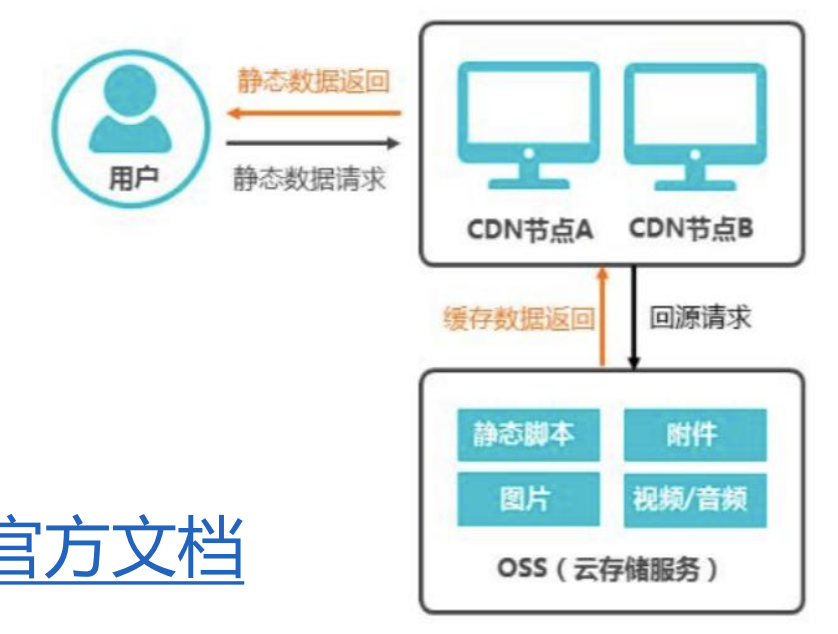
#### 什么是CDN ？

CDN是接近用户地理位置的边缘服务器，可以作为缓存处理



远距离访问文件服务器使用 CDN 技术进行优化

 阿里云CDN官方文档



## 4. 使用页面静态化技术减轻服务器负担

### 页面静态化

#### 1. 问题分析

现在，我们的页面是通过Thymeleaf模板引擎渲染后返回到客户端。在后台需要大量的数据查询，而VX：study322 其他均为翻录倒卖

后渲染得到HTML页面。会对数据库造成压力，并且请求的响应时间过长，并发能力不高。大家能想到什么办法来解决这个问题？

首先我们能想到的就是缓存技术，比如之前学习过的Redis。不过Redis适合数据规模比较小的情况。假如数据量比较大，例如我们的商品详情页。每个页面如果10kb，100万商品，就是10GB空间，对内存占用比较大。此时就给缓存系统带来极大压力，如果缓存崩溃，接下来倒霉的就是数据库了。

所以缓存并不是万能的，某些场景需要其它技术来解决，比如静态化。

#### 2. 什么是静态化？

静态化是指把动态生成的HTML页面变为静态内容保存，以后用户的请求到来，直接访问静态页面，不再经过服务的渲染。

而静态的HTML页面可以部署在nginx中，从而大大提高并发能力，减小tomcat压力。

#### 3. 如何实现静态化？

目前，静态化页面都是通过模板引擎来生成，而后保存到nginx服务器来部署。常用的模板引擎比如：• Freemarker

• Velocity

• Thymeleaf

我们之前就使用的Thymeleaf，来渲染html返回给用户。Thymeleaf除了可以把渲染结果写入Response，也可以写到本地文件，从而实现静态化。

#### 4.具体实现

1.创建静态页面service开发

2. 页面测试

3. 生成了静态页面