

# 第六章 系统优化项之页面技术

欧阳修





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

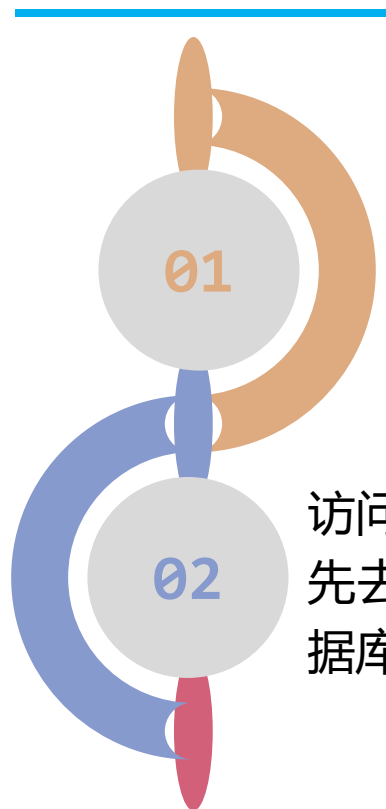


全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

# 为什么需要缓存预热？

### 1. 缓存预热步骤分析

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，  
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



### 1.2.1 秒杀商品详情改造

#### 缓存预热代码实现

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

```
/**
 * 秒杀商品详情
 *
 * @param resultMap
 * @param seckillActivityId
 * @return
 */
@RequestMapping("/seckill/detail/{seckillActivityId}")
public String itemPage(Map<String, Object> resultMap, @PathVariable long seckillActivityId) {
    SeckillActivity seckillActivity;
    SeckillCommodity seckillCommodity;

    String seckillActivityInfo = redisService.getValue( key: "seckillActivity:" + seckillActivityId);
    if (StringUtils.isEmpty(seckillActivityInfo)) {
        log.info("redis缓存数据:" + seckillActivityInfo);
        seckillActivity = JSON.parseObject(seckillActivityInfo, SeckillActivity.class);
    } else {
        seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);
    }
}
```

```
String seckillCommodityInfo = redisService.getValue( key: "seckillCommodity:" + seckillActivity.getCommodityId());
if (StringUtils.isEmpty(seckillCommodityInfo)) {
    log.info("redis缓存数据:" + seckillCommodityInfo);
    seckillCommodity = JSON.parseObject(seckillActivityInfo, SeckillCommodity.class);
} else {
    seckillCommodity = seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());
}

resultMap.put("seckillActivity", seckillActivity);
resultMap.put("seckillCommodity", seckillCommodity);
resultMap.put("seckillPrice", seckillActivity.getSeckillPrice());
resultMap.put("oldPrice", seckillActivity.getOldPrice());
resultMap.put("commodityId", seckillActivity.getCommodityId());
resultMap.put("commodityName", seckillCommodity.getCommodityName());
resultMap.put("commodityDesc", seckillCommodity.getCommodityDesc());
return "seckill_detail";
}
```

### 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

```
/**
 * 将秒杀详情相关信息倒入redis
 * @param seckillActivityId
 */
public void pushSeckillInfoToRedis(long seckillActivityId) {
    SeckillActivity seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);
    redisService.setValue("seckillActivity:" + seckillActivityId, JSON.toJSONString(seckillActivity));

    SeckillCommodity seckillCommodity = seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());
    redisService.setValue("seckillCommodity:" + seckillActivity.getCommodityId(), JSON.toJSONString(seckillCommodity));
}
```



### 1.2.3 执行测试

#### 缓存预热代码实现

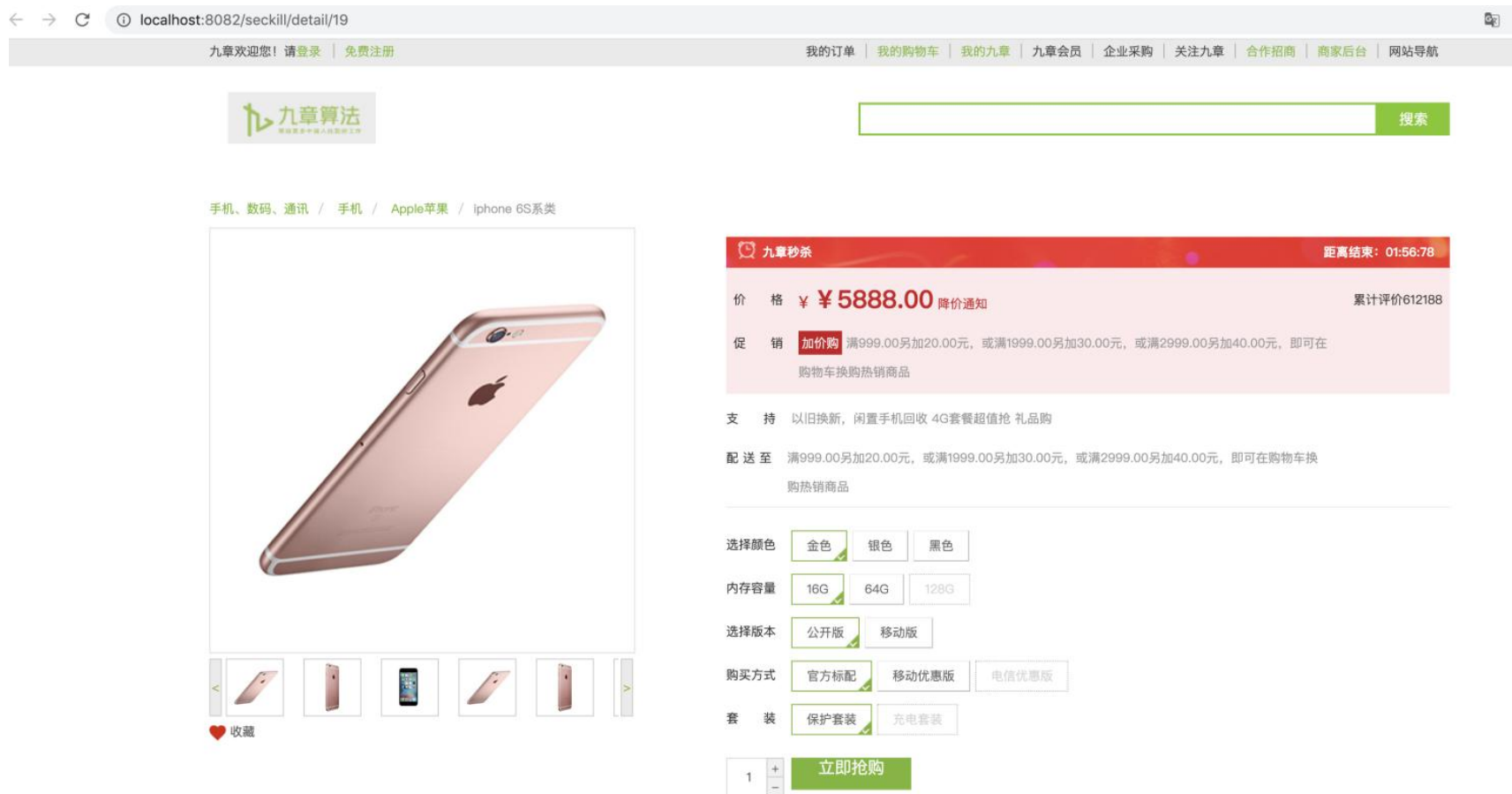
```
@Test
public void pushSeckillInfoToRedisTest(){
    seckillActivityService.pushSeckillInfoToRedis( seckillActivityId: 19);
}
```

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有



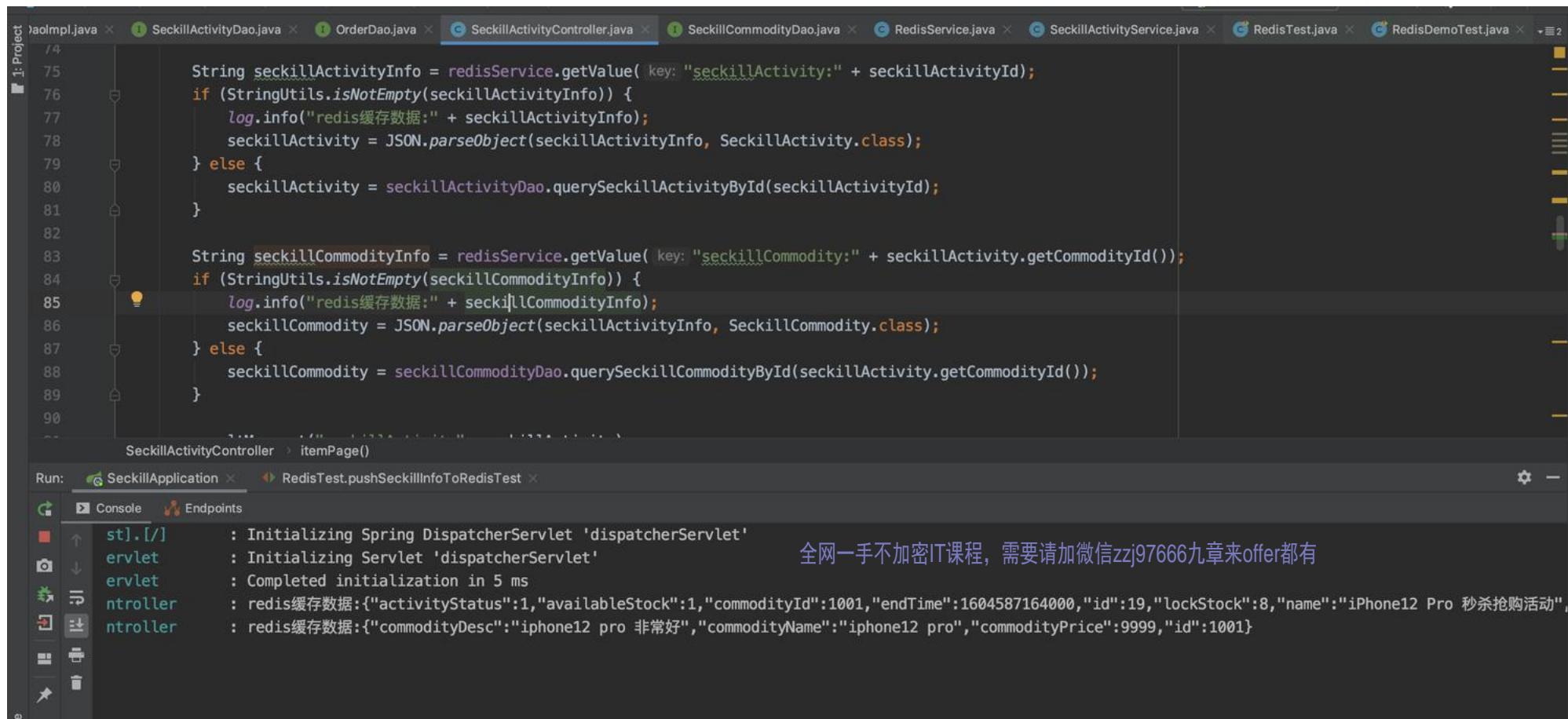
### 1.2.4 测试页面访问

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



```
String seckillActivityInfo = redisService.getValue( key: "seckillActivity:" + seckillActivityId);
if (StringUtils.isEmpty(seckillActivityInfo)) {
    log.info("redis缓存数据:" + seckillActivityInfo);
    seckillActivity = JSON.parseObject(seckillActivityInfo, SeckillActivity.class);
} else {
    seckillActivity = seckillActivityDao.querySeckillActivityById(seckillActivityId);
}

String seckillCommodityInfo = redisService.getValue( key: "seckillCommodity:" + seckillActivity.getCommodityId());
if (StringUtils.isEmpty(seckillCommodityInfo)) {
    log.info("redis缓存数据:" + seckillCommodityInfo);
    seckillCommodity = JSON.parseObject(seckillCommodityInfo, SeckillCommodity.class);
} else {
    seckillCommodity = seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());
}
```

Run: SeckillApplication x RedisTest.pushSeckillInfoToRedisTest x

Console

```
st.[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
ervlet : Initializing Servlet 'dispatcherServlet'
ervlet : Completed initialization in 5 ms
ntroller : redis缓存数据:{"activityStatus":1,"availableStock":1,"commodityId":1001,"endTime":1604587164000,"id":19,"lockStock":8,"name":"iPhone12 Pro 秒杀抢购活动",
ntroller : redis缓存数据:{"commodityDesc":"iphone12 pro 非常好","commodityName":"iphone12 pro","commodityPrice":9999,"id":1001}
```

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

## 2. 同步抢购倒计时



### 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  
一

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  
二

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  
三

前端轮询(Poll)服务器的时间，并获取距离活动开始的时间差

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

前端页面

循环地获取并校准时钟

服务端

### 后端添加一个接口向前端提供时间

```
/**
 * 获取当前服务器端的时间
 * @return
 */
@ResponseBody
@RequestMapping("/seckill/getSystemTime")
public String getSystemTime() {
    // 设置日期格式
    SimpleDateFormat df = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm:ss");
    // new Date() 为获取当前系统时间
    String date = df.format(new Date());
    return date;
}
```



# 3. 使用 CDN 技术进行优化

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

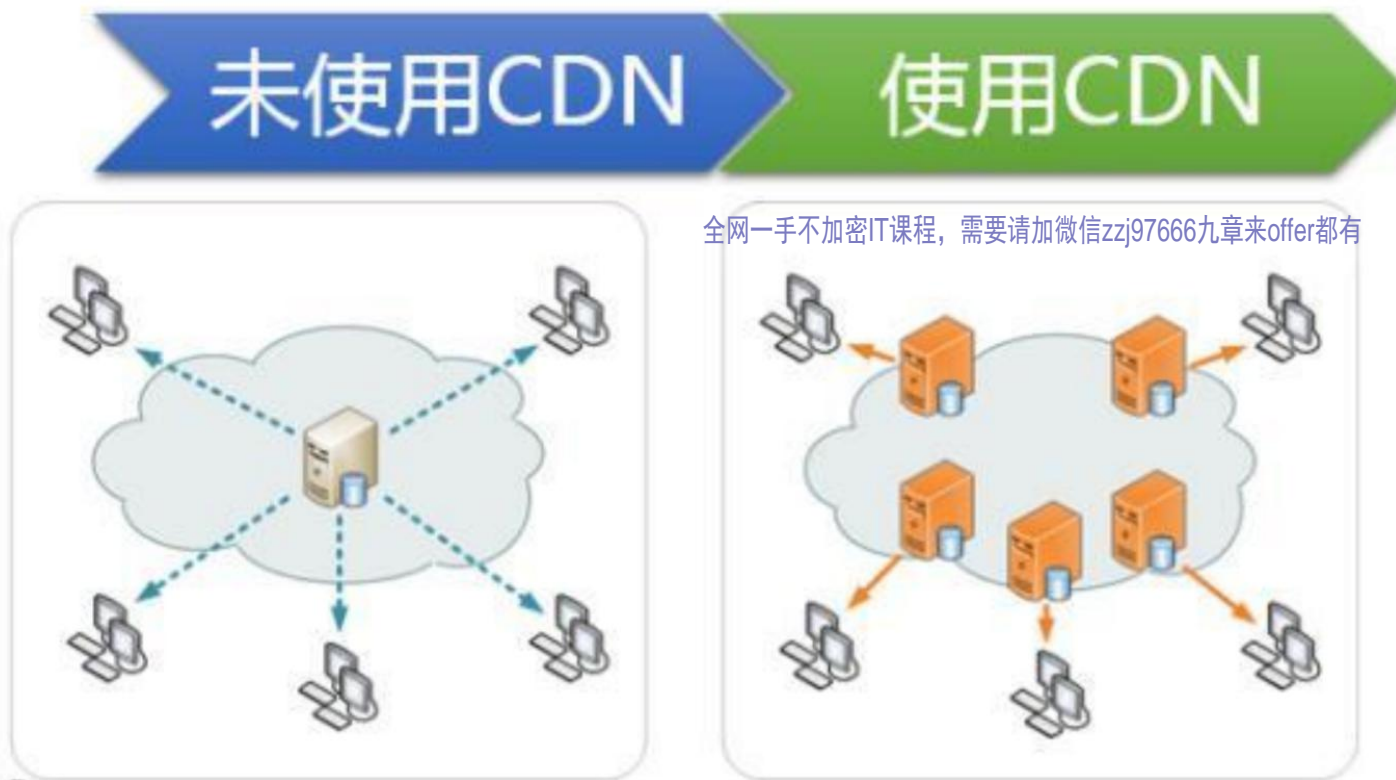
CDN (Content Delivery Network)

# 为什么要使用 CDN 技术?

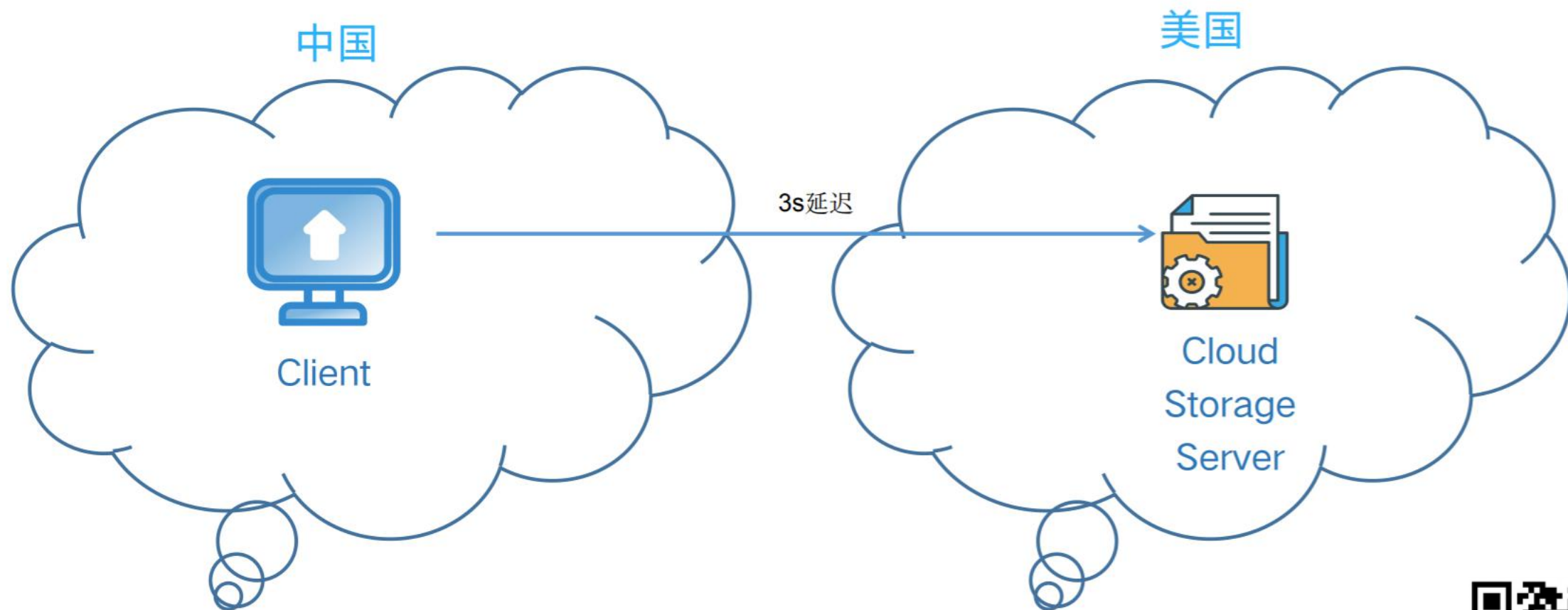


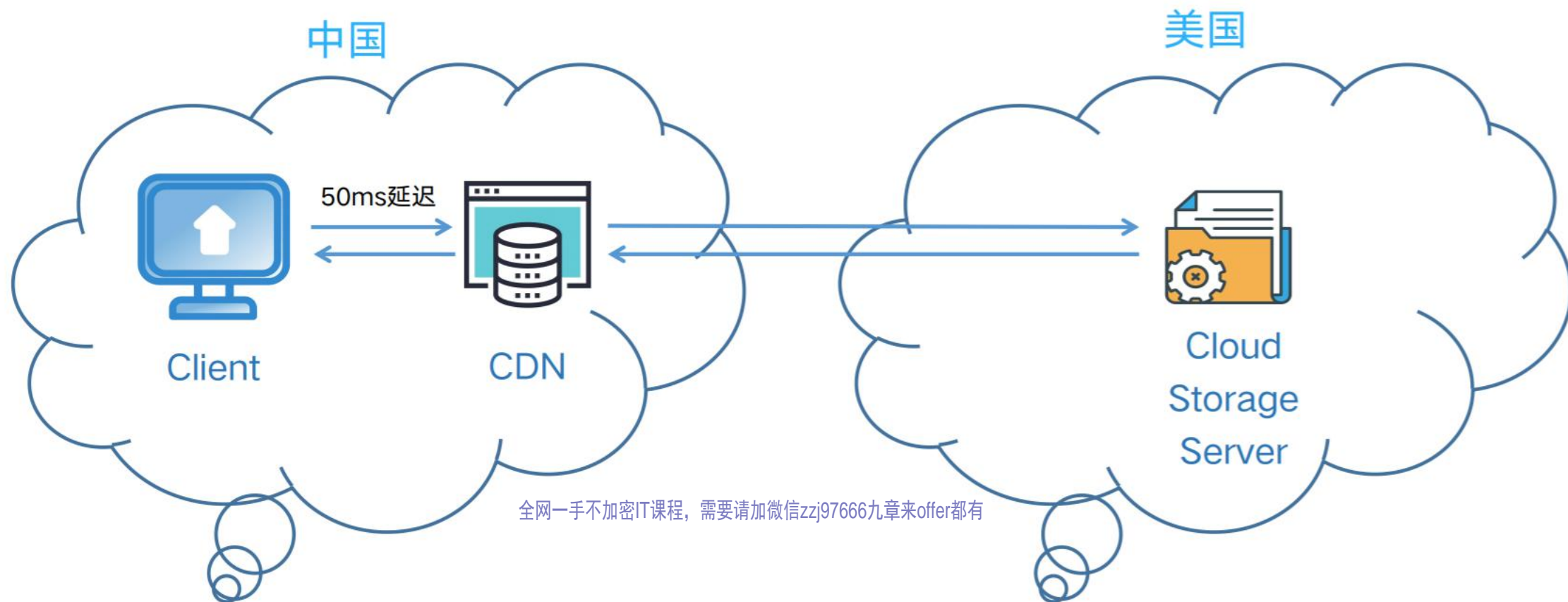
## 什么是 CDN ?

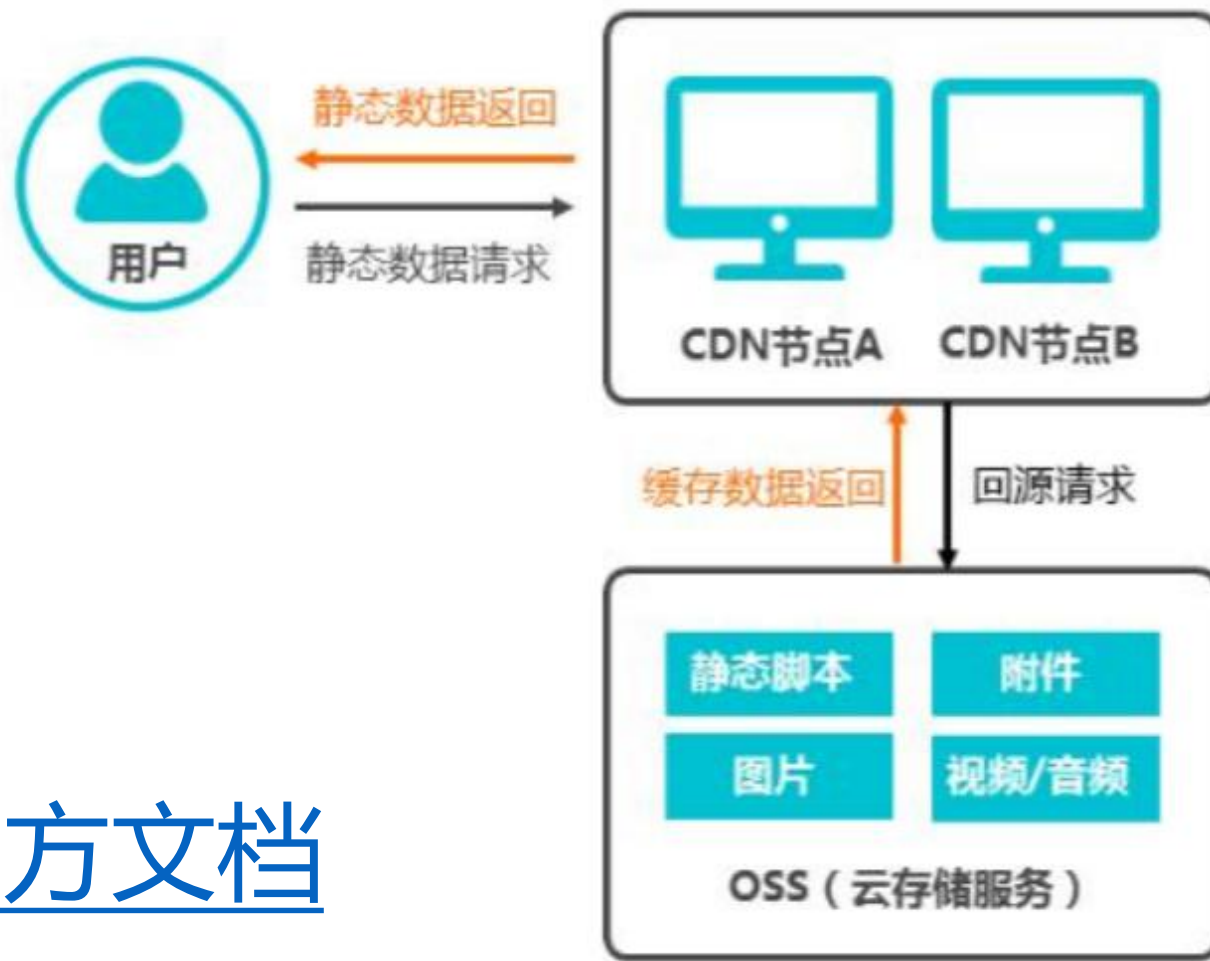
CDN是接近用户地理位置的边缘服务器，可以作为缓存处理











## 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

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

## 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。

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

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

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

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

```
public void createActivityHtml(Long seckillActivityId) {  
    PrintWriter writer = null;  
    try {  
        SeckillActivity seckillActivity =  
            seckillActivityDao.querySeckillActivityById(seckillActivityId);  
        SeckillCommodity seckillCommodity =  
            seckillCommodityDao.querySeckillCommodityById(seckillActivity.getCommodityId());  
        // 获取页面数据  
        Map<String, Object> resultMap = new HashMap<>();  
        resultMap.put("seckillActivity", seckillActivity);  
        resultMap.put("seckillCommodity", seckillCommodity);  
        resultMap.put("seckillPrice", seckillActivity.getSeckillPrice());  
        resultMap.put("oldPrice", seckillActivity.getOldPrice());  
        resultMap.put("commodityId", seckillActivity.getCommodityId());  
        resultMap.put("commodityName", seckillCommodity.getCommodityName());  
        resultMap.put("commodityDesc", seckillCommodity.getCommodityDesc());  
    }  
}
```

```
// 创建thymeleaf上下文对象  
Context context = new Context();  
// 把数据放入上下文对象  
context.setVariables(resultMap);  
  
// 创建输出流  
File file = new File(pathname: "/Users/mtdp /jiuZhangCode/seckill/src/main/resources/templates/" +  
    "seckill_detail_" + seckillActivityId + ".html");  
writer = new PrintWriter(file);  
// 执行页面静态化方法  
templateEngine.process(template: "seckill_detail", context, writer);  
} catch (Exception e) {  
    log.error(e.toString());  
    log.error("页面静态化出错: " + seckillActivityId);  
} finally {  
    if (writer != null) {  
        writer.close();  
    }  
}
```





## 4. 具体实现

### 2. 页面测试

```
@SpringBootTest
@RunWith(SpringRunner.class)
public class ThymeleafServiceTest {

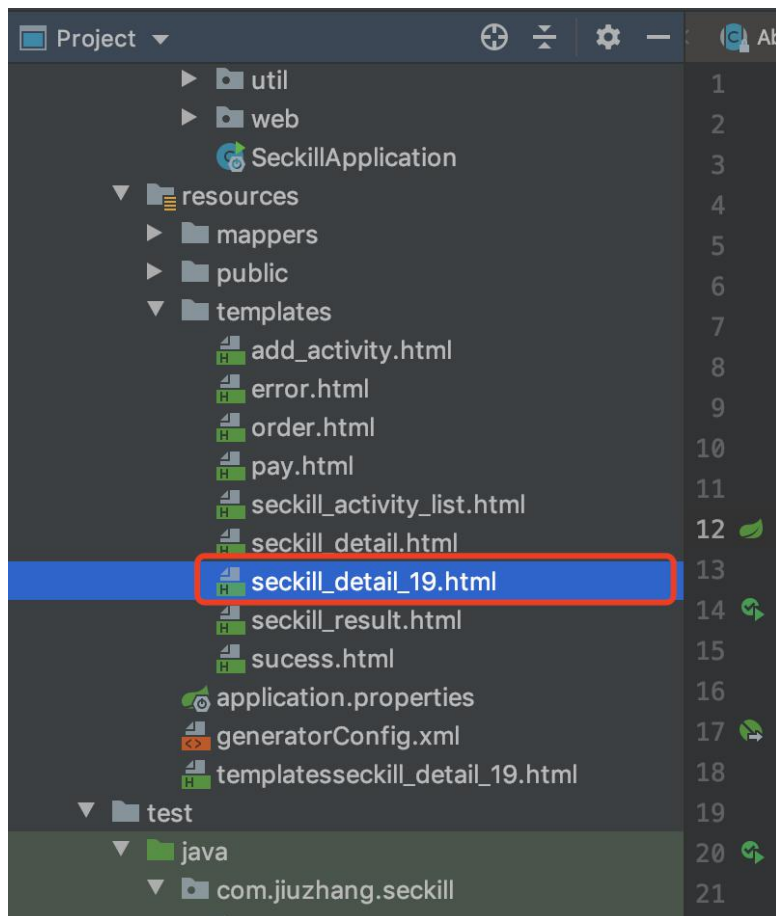
    @Autowired
    private ActivityHtmlPageService activityHtmlPageService;

    @Test
    public void createHtmlTest(){
        activityHtmlPageService.createActivityHtml( seckillActivityId: 19);
    }
}
```

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

## 4. 具体实现

### 3. 生成了静态页面



# 回顾与总结

全网一手不加密IT课程，需要请加微信zzj97666九章来offer都有

回顾并总结本节主要的知识点



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

