

二

41 案例：Redis 问题汇总和相关解决方案

本文收集了一些 Redis 使用中经常遇到的一些问题，和与之相对应的解决方案，这些内容不但会出现在实际工作中，也是面试的高频问题，接下来一起来看。

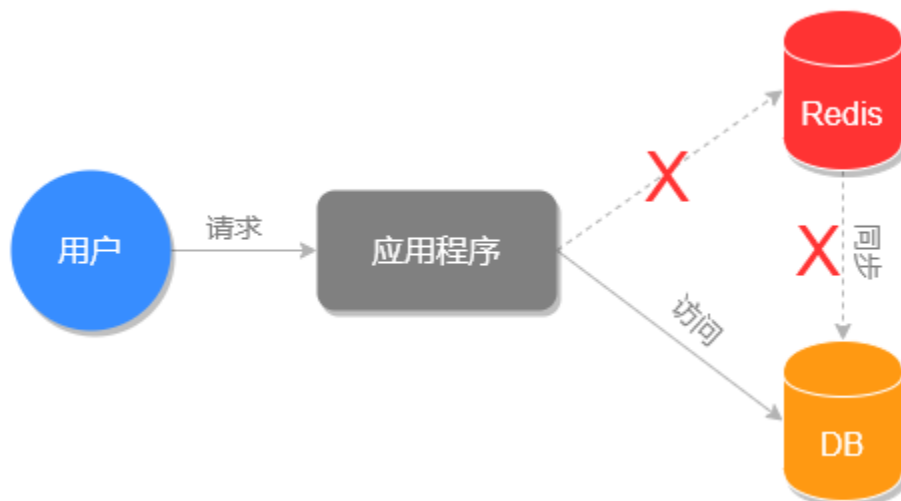
缓存雪崩

缓存雪崩是指在短时间内，有大量缓存同时过期，导致大量的请求直接查询数据库，从而对数据库造成了巨大的压力，严重情况下可能会导致数据库宕机的情况叫做缓存雪崩。

我们先来看下正常情况下和缓存雪崩时程序的执行流程图，正常情况下系统的执行流程如下图所示：



缓存雪崩的执行流程，如下图所示：



以上对比图可以看出缓存雪崩对系统造成的影响，那如何解决缓存雪崩的问题？

缓存雪崩的**常用解决方案**有以下几个。

加锁排队

加锁排队可以起到缓冲的作用，防止大量的请求同时操作数据库，但它的缺点是增加了系统的响应时间，降低了系统的吞吐量，牺牲了一部分用户体验。

加锁排队的代码实现，如下所示：

```
// 缓存 key
String cacheKey = "userlist";
// 查询缓存
String data = jedis.get(cacheKey);
if (StringUtils.isNotBlank(data)) {
    // 查询到数据，直接返回结果
    return data;
} else {
    // 先排队查询数据库，在放入缓存
    synchronized (cacheKey) {
        data = jedis.get(cacheKey);
        if (!StringUtils.isNotBlank(data)) { // 双重判断
            // 查询数据库
            data = findUserInfo();
            // 放入缓存
            jedis.set(cacheKey, data);
        }
        return data;
    }
}
```

以上为加锁排队的实现示例，读者可根据自己的实际项目情况做相应的修改。

随机化过期时间

为了避免缓存同时过期，可在设置缓存时添加随机时间，这样就可以极大的避免大量的缓存同时失效。

示例代码如下：

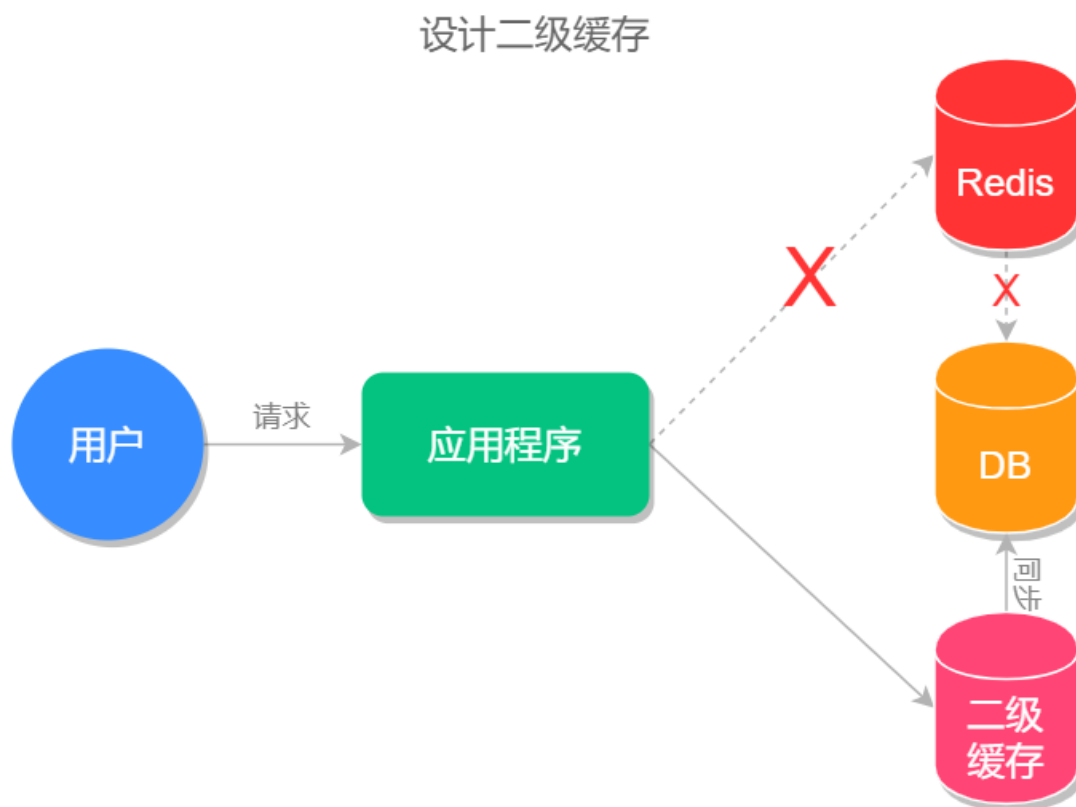
```
// 缓存原本的失效时间
int exTime = 10 * 60;
// 随机数生成类
Random random = new Random();
// 缓存设置
jedis.setex(cacheKey, exTime+random.nextInt(1000) , value);
```

设置二级缓存

二级缓存指的是除了 Redis 本身的缓存, 再设置一层缓存, 当 Redis 失效之后, 先去查询二级缓存。

例如可以设置一个本地缓存, 在 Redis 缓存失效的时候先去查询本地缓存而非查询数据库。

加入二级缓存之后程序执行流程, 如下图所示:

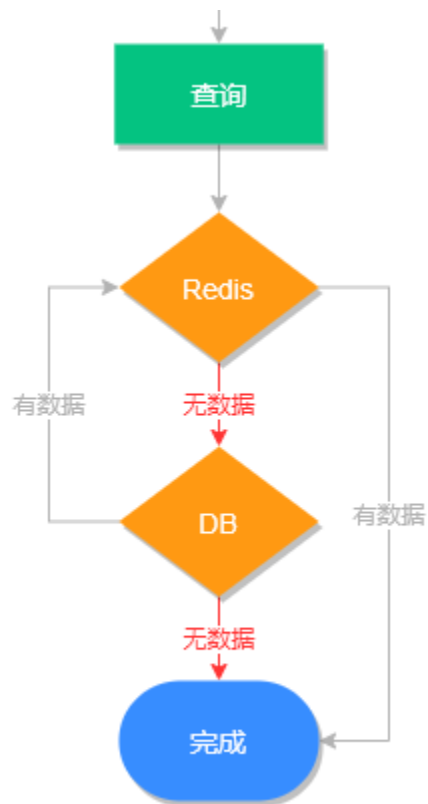


缓存穿透

缓存穿透是指查询数据库和缓存都无数据, 因为数据库查询无数据, 出于容错考虑, 不会将结果保存到缓存中, 因此每次请求都会去查询数据库, 这种情况就叫做缓存穿透。

缓存穿透执行流程如下图所示:





其中红色路径表示缓存穿透的执行路径，可以看出缓存穿透会给数据库造成很大的压力。

缓存穿透的解决方案有以下几个。

使用过滤器

我们可以使用过滤器来减少对数据库的请求，例如使用我们前面章节所学的布隆过滤器，我们这里简单复习一下布隆过滤器，它的原理是将数据库的数据哈希到 bitmap 中，每次查询之前，先使用布隆过滤器过滤掉一定不存在的无效请求，从而避免了无效请求给数据库带来的查询压力。

缓存空结果

另一种方式是我们可以把每次从数据库查询的数据都保存到缓存中，为了提高前台用户的使用体验 (解决长时间内查询不到任何信息的情况)，我们可以将空结果的缓存时间设置得短一些，例如 3~5 分钟。

缓存击穿

缓存击穿指的是某个热点缓存，在某一时刻恰好失效了，然后此时刚好有大量的并发请求，此时这些请求将会给数据库造成巨大的压力，这种情况就叫做缓存击穿。

缓存击穿的执行流程如下图所示：



它的解决方案有以下 2 个。

加锁排队

此处理方式和缓存雪崩加锁排队的方法类似，都是在查询数据库时加锁排队，缓冲操作请求以此来减少服务器的运行压力。

设置永不过期

对于某些热点缓存，我们可以设置永不过期，这样就能保证缓存的稳定性，但需要注意在数据更改之后，要及时更新此热点缓存，不然就会造成查询结果的误差。

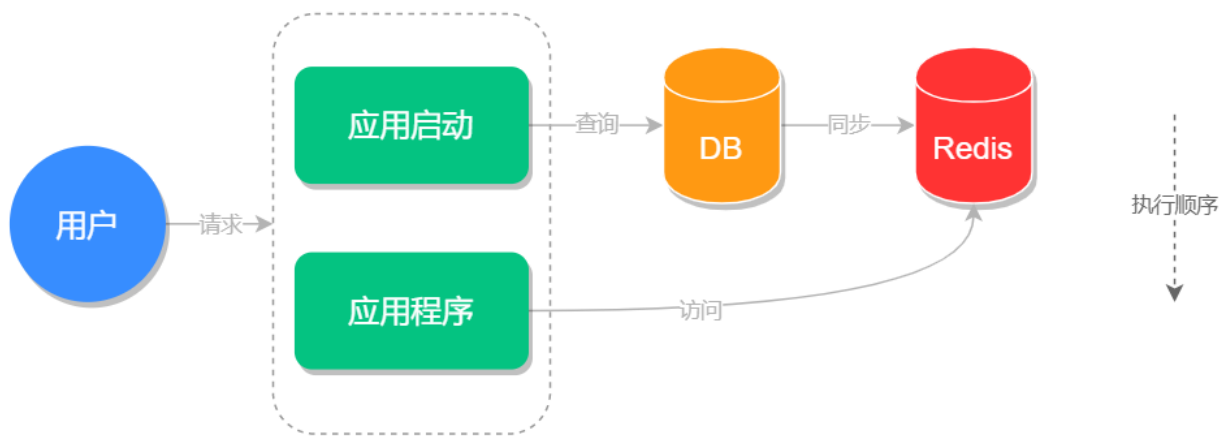
缓存预热

首先来说，缓存预热并不是一个问题，而是使用缓存时的一个优化方案，它可以提高前台用户的使用体验。

缓存预热指的是在系统启动的时候，先把查询结果预存到缓存中，以使用户后面查询时可以直接从缓存中读取，以节约用户的等待时间。

缓存预热的执行流程，如下图所示：

缓存预热



缓存预热的实现思路有以下三种：

1. 把需要缓存的方法写在系统初始化的方法中，这样系统在启动的时候就会自动的加载数据并缓存数据；
2. 把需要缓存的方法挂载到某个页面或后端接口上，手动触发缓存预热；
3. 设置定时任务，定时自动进行缓存预热。

小结

本文介绍了缓存雪崩产生的原因是因为短时间内大量缓存同时失效，而导致大量请求直接查询数据库的情况，解决方案是加锁、随机设置过期时间和设置二级缓存等；还介绍了查询数据库无数据时会导致的每次空查询都不走缓存的缓存穿透问题，解决方案是使用布隆过滤器和缓存空结果等；同时还介绍了缓存在某一个高并发时刻突然失效导致的缓存击穿问题，以及解决方案——加锁、设置永不过期等方案，最后还介绍了优化系统性能的手段缓存预热。

[上一页](#)

[下一页](#)