【第45话：缓存穿透、缓存击穿、缓存雪崩听着挺吓人，回答起来还是很容易的】

Hello 小伙伴们，这节课我们说一下高频面试题：“缓存穿透、缓存击穿、缓存雪崩是什么，如何防止”

在目前互联网项目中缓存工具可以说是必不可少的。在Java项目中Redis是目前使用最多的缓存工具。使用缓存工具最主要的目的就是提高查询性能、保护数据库服务器。但是随着Redis的时候，在一些极端情况下可能导致缓存失效，这时所有的请求一下都访问到数据库服务器上了，可能导致数据库服务器瞬间宕机，这时DBA即使重启服务器，也可能出现服务器被新来的请求导致再次挂掉。

在几年前，国内的一个比较知名的互联网公司就因为缓存雪崩，整个项目的后台服务器全部挂掉，事故从当天下午一直持续到凌晨3~4点，导致公司直接经济损失几千万。对于这家公司当时的情况，完全可以拍一个电影《生死十二小时》。所以为了防止公司出现巨大损失，面试官都会要求我们具备防止出现这类事情的能力。面试过程中这三个问题可以说出现的频率是非常之高的。

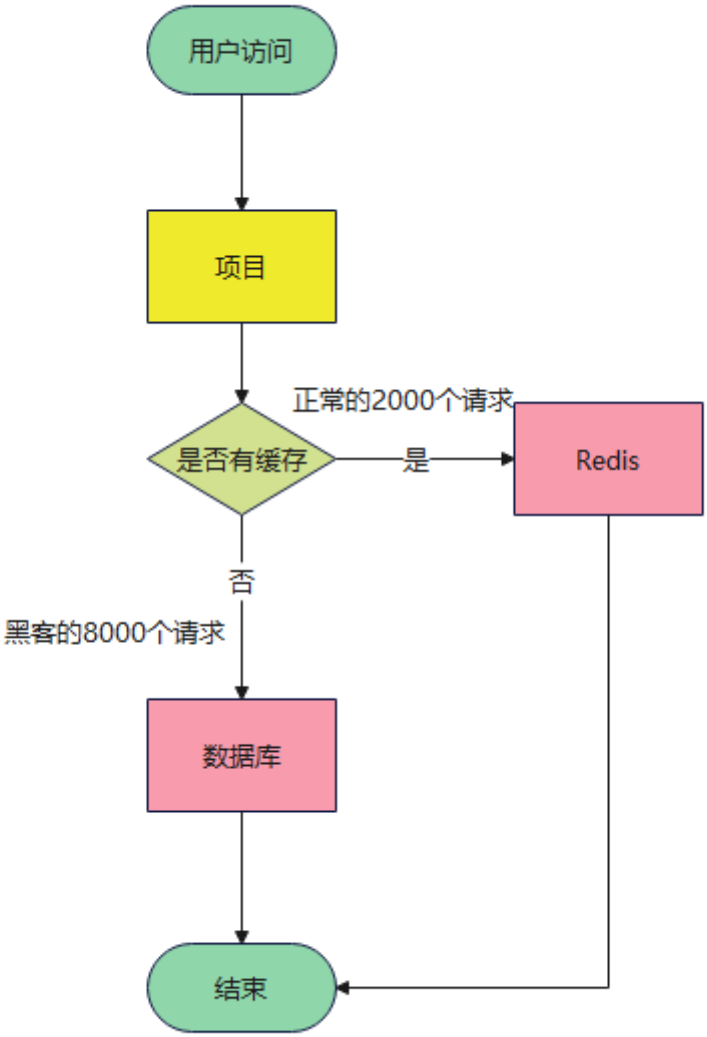
下面分别给各位小伙伴们说明一下缓存穿透、缓存击穿、缓存雪崩是如何出现的，以及解决方案。

在分别说这三种情况之前，小伙伴们要有一个总体概念：这些情况都是绕过了Redis直接访问数据库。他们的区别就是如何绕过Redis的。

缓存穿透

缓存穿透绝大多数都是出现在非法访问的环境下。例如：一秒有1万个请求，其中8000个请求是黑客发起的攻击。这8000个请求在Redis中没有，在数据库中也没有。

一个具体的例子：在使用数据库时绝大多数主键都是从1开始的。这时黑客请求ID=-1的数据，Redis中没有这个数据，数据库中也没有这个数据。就会出现访问都走了Redis和数据库，而且访问过后也不会缓存，后面的几千次、几万次访问都走数据库，这样一下就把数据库压垮了。这种情况就是缓存穿透。



缓存穿透解决办法也很简单：

（1）设置有效时间。访问数据库后，即使查询到的是null，也进行缓存。但可以对null的数据缓存的时间相对短一些。例如：正常查询到的结果缓存3天，而null值数据缓存6小时。

（2）使用布隆过滤器。对数据库中的数据进行标记，如果缓存不存在时，先使用布隆过滤器判断数据库中是否存在这个数据，如果不存在不再访问数据库， 直接把null值缓存到Redis中。

缓存击穿

缓存击穿多出现在热点数据。热点数据意味着数据访问频率比较高。例如手机搜索时华为手机可能就是热点数据。存在Redis的数据我们都会设置个有效期，在高并发下，大量请求来了，恰巧在Redis中这个数据的有效期失效了，这些请求就直接去访问数据库了。数据库可能又挂掉了，可怜了我的数据库，没有保护时怎么这么脆弱。这就像Redis作为一个盾牌，保护着数据库，但因为盾牌的一个点被击穿了，而导致数据库的保护失效。这种情况被称为缓存击穿。

防止缓存击穿的办法有很多种：

（1）永久数据。若缓存数据不会发生变更，可尝试永久数据。

（2）加锁。如果缓存数据更新不频繁，可以考虑加锁。如果是单体架构项目可以考虑添加本地锁，例如：ReentrantLock。如果是分布式项目可结合Zookeeper或Redis添加分布式锁。

例如：使用ReentrantLock添加本地锁。

```
private ReentrantLock lock = new ReentrantLock();
@Override
public Item selectByid(Integer id) {
    lock.lock();
    String key = "item:"+id;
    if(redisTemplate.hasKey(key)){
        return (Item) redisTemplate.opsForValue().get(key);
    }
    if(lock.isLocked()) {
        Item item = itemDubboService.selectById(id);
        // 由于设置了有效时间，就可能出现缓存击穿问题
        redisTemplate.opsForValue().set(key, item, 7, TimeUnit.DAYS);
        lock.unlock();
        return item;
    }
    // 如果加锁失败，为了保护数据库，直接返回null
    return null;
}
```

（3）定时更新。如果缓存数据更新频繁，可考虑定时更新或双写一致。例如在修改MySQL数据库数据时，同步更新Redis的缓存数据，同时重置缓存失效时间。也可以使用定时任务，周期性检查缓存失效时间，如果发现缓存快失效，延长缓存失效时间。

缓存雪崩

缓存雪崩发生在一段时间内大量缓存失效或Redis服务器全盘宕机的情况下。和缓存击穿区别很明显，缓存击穿是因为一点失效而导致的保护失效。而雪崩是大面试失效。

我们先感受下自然现象中雪崩的感觉

想要尽量避免缓存雪崩的办法：

（1）Redis搭建高可用集群。使用主从复制+哨兵，避免Redis全盘宕机。

（2）永久数据。如果服务器内存充足，不是海量缓存数据，可考虑使用永久数据避免同一时间段，大量缓存失效。

（3）限流。可使用Hystrix降级或限流，限制入口流量。也可以使用Gateway的限流。这样虽然一些请求被降级了，用户访问不到真实资源。但是用户再次刷新后和其他流量错时请求，最终也可以访问到真实资源的。

（4）自定义有效时间算法。对于一些数据可能都是在同一时间段进行的缓存。一种比较简单的办法是，缓存时添加点随机时间。把缓存时间尽量错开。例如：正常缓存都是7天时间，每次缓存时在后面添加个随机数。

```
int seconds = random.nextInt(10000);
redisTemplate.opsForValue().set(key, item, 604800+ seconds, TimeUnit.SECONDS);
```

按照刚刚我讲的内容给面试官回答，这个问题还是很简单的吧。