#### 1、什么是缓存雪崩？怎么解决？



通常，我们会使用缓存用于缓冲对DB的冲击，如果缓存宕机，所有请求将直接打在DB，造成DB宕机——从而导致整个系统宕机。

两种解决策略（同时使用）：



* 对缓存做高可用，防止宕机
* 使用断路器，如果缓存宕机，为了防止系统全部宕机，限制部分流量进入DB，保证部分可用，其余的请求返回断路器的默认值。

#### 2、什么是缓存穿透？怎么解决？

解释1：缓存查询一个没有的key，同时数据库也没有，如果黑客大量使用这种方式，那么就会导致DB宕机。

解决方案：可以使用一个默认值来防止，例如，当访问一个不存在的key，然后再去访问数据库，还是没有，那么就在缓存里放一个占位符，下次来的时候，检查这个占位符，如果发生时占位符，就不去数据库查询了，防止DB宕机。

解释2：大量请求查询一个刚刚失效的key，导致DB压力倍增，可能导致宕机，但实际上，查询的都是相同的数据。

解决方案：可以在这些请求代码上加上双重检查锁。但是那个阶段的请求会变慢。不过总比DB宕机好。

#### 3、什么是缓存并发竞争？怎么解决？

解释：多个客户端写一个key，如果顺序错了，数据就不对了。但是顺序我们无法控制。

解决方案：使用分布式锁，例如zk，同时加入数据的时间戳。同一时刻，只有抢到锁的客户端才能写入，同时，写入时，比较当前数据的时间戳和缓存中数据的时间戳。

#### 4、什么是缓存和数据库双写不一致？怎么解决？

解释：连续写数据和缓存，但是操作期间，出现并发了。数据不一致了。

通常，更新缓存和数据库有以下几种顺序：

* 先更新数据库，再更新缓存。
* 先删缓存，再更新数据库。
* 先更新数据库，再删除缓存。

三种方式的优劣来看一下：

1. 先更新数据库，再更新缓存。

这么做的问题是：当有2个请求同时更新数据，那么如果不使用分布式锁，将无法控制最后缓存的值到底是多少。也就是并发写的时候有问题。

1. 先删缓存，再更新数据库

这么做的问题：如果在删除缓存后，有客户端读数据，将可能读到旧数据，并可能设置到缓存中，导致缓存中的数据一直是老数据。

有2种解决方案：

使用“双删”，即删更删，最后一步的删除作为异步操作，就是防止有客户端读取的时候设置了旧值。

使用队列，当这个 key 不存在时，将其放入队列，串行执行，必须等到更新数据库完毕才能读取数据。

1. 先更新数据库，再删除缓存

这个实际是常用的方案，但是有很多人不知道，这里介绍以下，这个叫Cache Aside Pattern。如果先更新数据库，再删除缓存，那么就会出现更新数据库之前有瞬间数据不是很及时。

同时，如果是在更新之前，缓存刚好失效了，度客户端有可能读到旧值，然后在客户端删除结束后再次设置了旧值，非常巧合的情况。

有2个前提条件：缓存在写之前的时候失效，同时，在写客户单删除操作结束后，放置旧数据——也局势读比写慢。设置有的写操作还会锁表。

所以，这个很难出现，但是如果出现了怎么办？使用双删！！！记录更新期间有没有客户端数据库，如果有，在更新完数据库之后，执行延迟删除。

还有一种可能，如果执行更新数据库，准备执行删除缓存时，服务挂了，执行删除

这就坑了！！！不过可以通过订阅数据库的binlog来删除。