

缓存穿透，击穿，雪崩

需要先了解一下这三个概念是什么。

缓存穿透：请求去查询一条数据库中根本不存在的数据库，这样每次请求都会打到数据库，这种查询不存在的数据现象就叫缓存穿透。

缓存击穿：在高并发的场景中，大量的请求同时查询一个key时，刚好这个key失效，就会导致大量的请求都打到数据库上，这种现象就称为缓存击穿。

缓存雪崩：缓存服务器宕机了，直接导致大量的请求打到数据库上，这种现象就称为缓存雪崩。

面对上面三种问题，我们有什么解决方案呢？

解决缓存穿透方案：

1. 缓存空值：由于缓存中没有存储这些空数据的key，那我们就为这些key存储一个null值到缓存中，这样后面再有请求过来，就会从缓存中取null值，注意不要忘了设置过期时间。
2. BloomFilter：布隆过滤器，通过K个hash函数将一个值计算出K个hash值，然后分别记录到这K个hash值对应的位数组上为1，当一个新的数据进来时，如果这个数据通过hash函数算出来的hash值对应的位数组的值都为1，表示该数据是存在的。所以我们可以利用布隆过滤器放到缓存前面，通过过滤器判断key是否存在，存在继续走缓存查询，如果不存在就直接返回系统不存在。

这两种方案都可以，第一种方案不适用大量的空数据存在的场景，第二种方案会有一定的误判率，因为布隆过滤器存在误判。

解决缓存击穿方案：

1. 当多个线程同时去查询数据库这条数据，那我们可以第一个查询的请求上使用一个**互斥锁**，来锁住它。其他线程拿不到锁就阻塞，等第一个线程完成查询后，将数据写入缓存后，后面的线程再进来就会从缓存中获取数据。这种方案由于会阻塞其他线程，导致系统吞吐量下降，需要结合具体业务处理。

解决缓存雪崩：

1. 事前：使用缓存集群，保证服务的高可用，如果是redis缓存可以使用主从+哨兵。
2. 事中：可以使用ehcache本地缓存。使用Hystrix进行限流或者降级，比如一秒钟有5000个请求过来，我们可以设置只能有一秒2000个请求通过这个组件，其他剩余的3000个请求走限流逻辑。然后就去调用我们自己开发的降级组件，比如设置一些默认值，保证数据库不会被大量请求怼爆炸。
3. 事后：如果是redis缓存，需要做持久化机制，保证数据能够最大化的恢复。

问答：

解决热点数据集中失效问题：对于一些热点数据来说，当缓存失效也会导致大量请求打到数据库，我们给出的方案是：热点数据设置的过期时间错开，比如在一个基础的时间上，做一个随机数的加减，让过期时间在基础时间上下做浮动。