缓存穿透

**1、概念**

缓存穿透的概念很简单，用户想要查询一个数据，发现redis内存数据库没有，也就是缓存没有命中，于是向持久层数据库查询。发现也没有，于是本次查询失败。当用户很多的时候，缓存都没有命中，于是都去请求了持久层数据库。这会给持久层数据库造成很大的压力，这时候就相当于出现了缓存穿透。

**这里需要注意和缓存击穿的区别，缓存击穿，是指一个key非常热点，在不停的扛着大并发，大并发集中对这一个点进行访问，当这个key在失效的瞬间，持续的大并发就穿破缓存，直接请求数据库，就像在一个屏障上凿开了一个洞。**

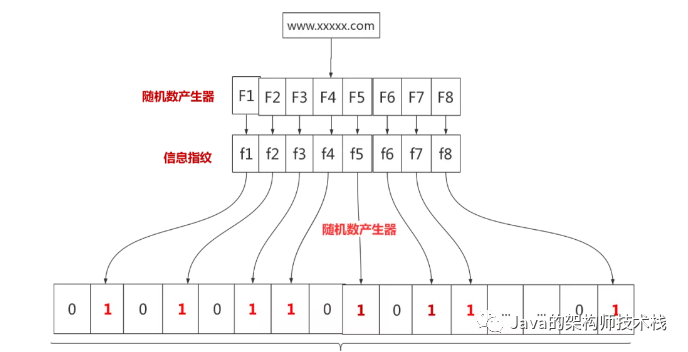
为了避免缓存穿透其实有很多种解决方案。下面介绍几种。

**2、解决方案**

**（1）布隆过滤器**

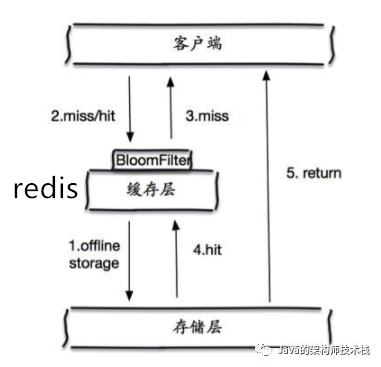
布隆过滤器是一种数据结构，垃圾网站和正常网站加起来全世界据统计也有几十亿个。网警要过滤这些垃圾网站，总不能到数据库里面一个一个去比较吧，这就可以使用布隆过滤器。假设我们存储一亿个垃圾网站地址。

可以先有一亿个二进制比特，然后网警用八个不同的随机数产生器（F1,F2, …,F8） 产生八个信息指纹（f1, f2, …, f8）。接下来用一个随机数产生器 G 把这八个信息指纹映射到 1 到1亿中的八个自然数 g1, g2, …,g8。最后把这八个位置的二进制全部设置为一。过程如下：



有一天网警查到了一个可疑的网站，想判断一下是否是XX网站，首先将可疑网站通过哈希映射到1亿个比特位数组上的8个点。如果8个点的其中有一个点不为1，则可以判断该元素一定不存在集合中。

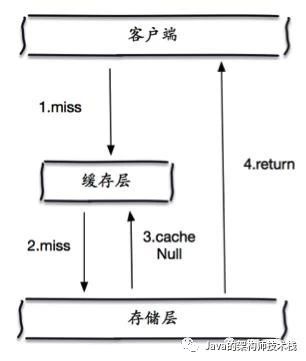
那这个布隆过滤器是如何解决redis中的缓存穿透呢？很简单首先也是对所有可能查询的参数以hash形式存储，当用户想要查询的时候，使用布隆过滤器发现不在集合中，就直接丢弃，不再对持久层查询。



这个形式很简单。

**2、缓存空对象**

当存储层不命中后，即使返回的空对象也将其缓存起来，同时会设置一个过期时间，之后再访问这个数据将会从缓存中获取，保护了后端数据源；



但是这种方法会存在两个问题：

1.如果空值能够被缓存起来，这就意味着缓存需要更多的空间存储更多的键，因为这当中可能会有很多的空值的键；

2.即使对空值设置了过期时间，还是会存在缓存层和存储层的数据会有一段时间窗口的不一致，这对于需要保持一致性的业务会有影响。