# 缓存穿透

## 参考资料

<https://blog.csdn.net/fei33423/article/details/79027790>

<https://blog.csdn.net/zeb_perfect/article/details/54135506>

## 定义

缓存穿透是指查询一个一定不存在的数据，由于缓存是不命中时需要从数据库查询，查不到数据则不写入缓存，这将导致这个不存在的数据每次请求都要到数据库去查询，造成缓存穿透。

## 解决方案

### Bloom filter

<https://diducoder.com/mass-data-topic-2-bloom-filter.html>

对所有可能查询的参数以hash形式存储，在控制层先进行校验，不符合则丢弃。还有最常见的则是采用布隆过滤器，将所有可能存在的数据哈希到一个足够大的bitmap中，一个一定不存在的数据会被这个bitmap拦截掉，从而避免了对底层存储系统的查询压力。

补充:

Bloom filter

适用范围：可以用来实现数据字典，进行数据的判重，或者集合求交集

基本原理及要点：对于原理来说很简单，位数组+k个独立hash函数。将hash函数对应的值的位数组置1，查找时如果发现所有hash函数对应位都是1说明存在，很明显这个过程并不保证查找的结果是100%正确的。同时也不支持删除一个已经插入的关键字，因为该关键字对应的位会牵动到其他的关键字。所以一个简单的改进就是counting Bloom filter，用一个counter数组代替位数组，就可以支持删除了。添加时增加计数器，删除时减少计数器。

如何根据输入元素个数n，确定位数组m的大小及hash函数个数。当hash函数个数k=(ln2)\*(m/n)时错误率最小。在错误率不大于E的情况 下，m至少要等于n\*lg(1/E)才能表示任意n个元素的集合。但m还应该更大些，因为还要保证bit数组里至少一半为0，则m应 该>=nlg(1/E)\*lge 大概就是nlg(1/E)1.44倍(lg表示以2为底的对数)。

### 缓存空对象. 将 null 变成一个值.

# 缓存雪崩

## 定义

缓存雪崩是指在我们设置缓存时采用了相同的过期时间，导致缓存在某一时刻同时失效，请求全部转发到DB，DB瞬时压力过重雪崩。

## 解决方案

### 原来的失效时间+随机时间

### 使用互斥锁(mutex key)

### "提前"使用互斥锁(mutex key)

### "永远不过期"

### 做二级缓存，或者双缓存策略。

### 加锁排队. 限流-- 限流算法. 1.计数 2.滑动窗口 3. 令牌桶Token Bucket 4.漏桶 leaky bucket

### 资源保护