# 缓存击穿：

## 解决方法：

### 设置并发锁

设置并发锁，防止请求大量请求数据库，如果获取到锁了，去数据库查询，如果没有，说明有其他线程在查询数据库，那么只需要重试一下就好了

public String get(key) {

String value = redis.get(key);

if (value == null) { //代表缓存值过期

//设置3min的超时，防止del操作失败的时候，下次缓存过期一直不能load db

if (redis.setnx(key\_mutex, 1, 3 \* 60) == 1) { //代表设置成功

value = db.get(key);

redis.set(key, value, expire\_secs);

redis.del(key\_mutex);

} else { //这个时候代表同时候的其他线程已经load db并回设到缓存了， // 这时候重试获取缓存值即可

Thread.sleep(50);

get(key); //重试

}

} else {

return value;

}

}

### 对空值缓存

如果一个查询数据为空（不管数据是否存在），都对该空结果进行缓存，其过期时间会设置非常短。

### 设置可以访问名单

使用bitmaps类型定义一个可以访问名单，名单id作为bitmaps的偏移量，每次访问时与bitmaps中的id进行比较，如果访问id不在bitmaps中，则进行拦截，不给其访问。

### 采用布隆过滤器

布隆过滤器可以判断元素是否存在集合中，他的优点是空间效率和查询时间都比一般算法快，缺点是有一定的误识别率和删除困难。

### 进行实时监控

对于redis缓存中命中率急速下降时，迅速排查访问对象和访问数据，将其设置为黑名单