



链滴

缓存穿透，缓存雪崩，缓存击透的概念及解决方案

作者：[Qiyue0726](#)

原文链接：<https://ld246.com/article/1564716032561>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



<senction>

<h1>缓存穿透</h1>

<p>当客户端频繁地向服务器请求一个缓存中不存在的数据时，服务器会不断查询数据库，数据库能会因此挂掉。</p>

<blockquote>解决方案</blockquote>

<p>这里有一个简单粗暴的方法。
当查询数据库时，如果结果返回null，这时仍然对其结果行缓存，但缓存的过期时间很短。</p>

</senction>

<senction>

<h1>缓存雪崩</h1>

<p>在设置缓存时都设置了相同的过期时间，致使缓存同时失效，所有请求将全部转发给数据库，数据库瞬时压力过重，导致雪崩。</p>

<blockquote>解决方案</blockquote>

<p>可以在原有的失效时间基础上增加一个随机值，使得每个key的过期时间分散开，不会集中在一段时间失效。</p>

</senction>

</senction>

<h1>缓存击穿</h1>

<p>一个存在的key，在缓存过期的那一刻，刚好有大量的请求，大并发的请求会击穿到DB，造成时DB请求量大，压力骤增。</p>

<p>这里的缓存击穿和缓存雪崩看似相同，实际上有一点区别。

* 击穿是针对的某一个key

● 雪崩是有很多的key同时失效

</p>

<blockquote>解决方案</blockquote>

<p>在访问key之前，可以采用SETNX (set if not exists) 来设置另一个短期的key，通过其来锁

当前key的访问，访问结束就删除该短期key。 </p>

```
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
            即可
                sleep(50);
                get(key); //重试
            }
        } else {
            return value;
        }
    }
}

</senction>
```