

【第49话：购物车商品放在Redis里，Redis里面放不下怎么办】

Hello 小伙伴们，这节课给大家讲解一个业务问题：“购物车商品放在Redis里，Redis里面放不下怎么办”。

这个问题主要从两面试进行回答：

- 购物车商品放在Redis里，Redis里面放不下怎么办
- 如果一个key放不下了怎么办

先来说说Redis内存满了的情况

我们在做电商类型项目时，会把购物车数据放入到Redis中。Redis虽然有持久化策略，但是运行过程中数据都存在于内存中。当用户量比较大、购物车数据量比较多时可能会出现Redis所占内存不足的情况。对于这样的情况可以考虑扩大服务器内存。但是服务器内存也是有上限的，这时就需要搭建Redis集群。即使搭建了集群最大节点个数为16384个，极限情况下也可能出现内存不足的情况。

这时就会需要Redis的缓存淘汰策略。在配置文件中有如下配置

```
# maxmemory-policy noeviction 默认策略noeviction
# maxmemory <bytes> 缓存最大阈值
# volatile-lru -> 在设置过期key集中选择使用数最小的。
# allkeys-lru -> 在所有key中选择使用最小的。
# volatile-lfu -> 在设置过期时间key集中采用lfu算法。
# allkeys-lfu -> 在所有key中采用lfu算法。
# volatile-random -> 在设置过期key集中随机删除。
# allkeys-random -> 在所有key中随机删除。
# volatile-ttl -> 在设置了过期时间key中删除最早过期时间的。
# noeviction -> 不删除key，超过时报错。
```

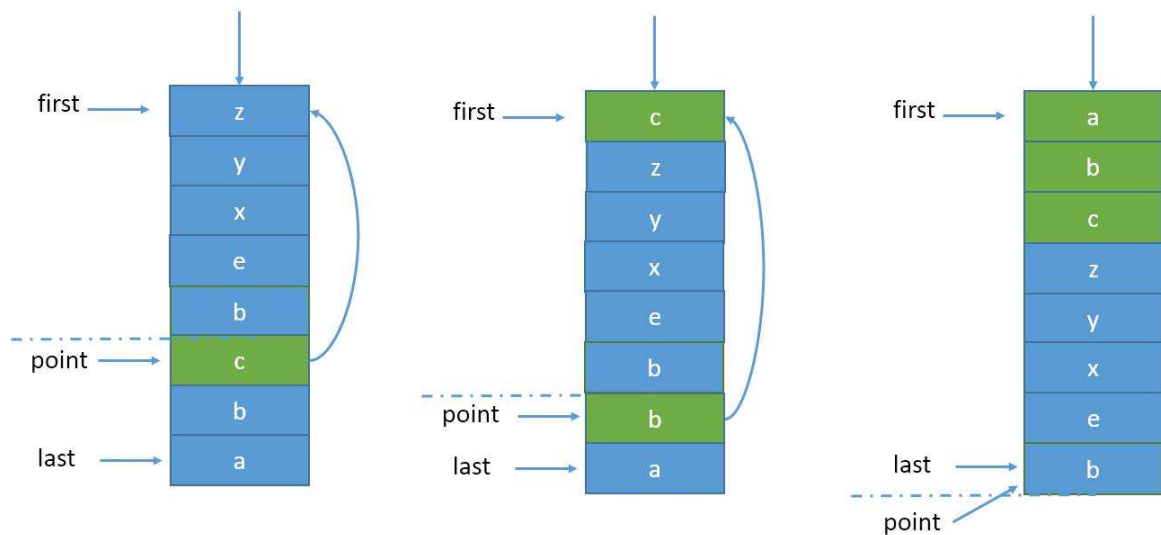
默认情况下是：Redis中数据炒作了最大内存阈值时使用noeviction策略，在新增时会报错。所以可以考虑使用其他几种淘汰策略。

为了能够更好的理解这些淘汰策略，先把几种算法搞到。

LRU

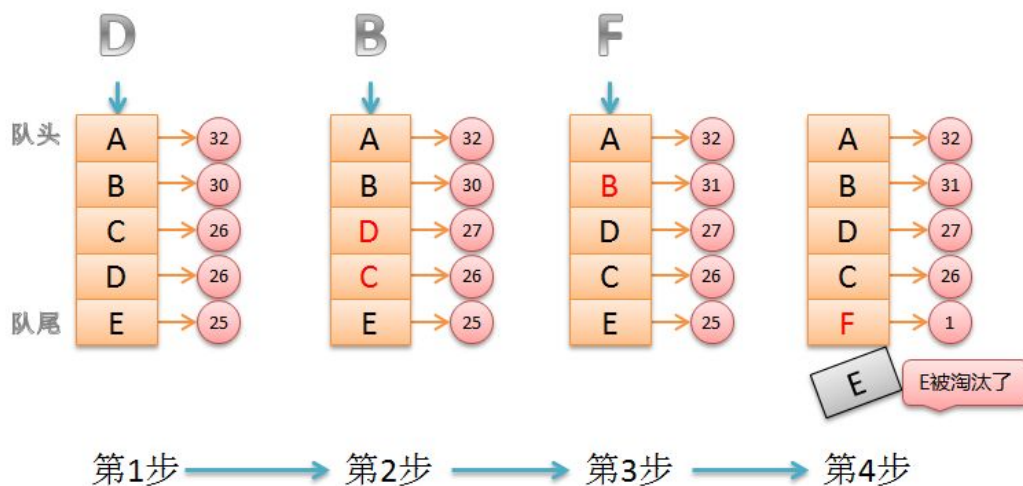
LRU (Least recently used) 最近最少使用，如果数据最近被访问过，那么将来被访问的几率也更高。LRU算法实现简单，运行时性能也良好，被广泛的使用在缓存/内存淘汰中。

1. 新数据插入到链表头部
2. 每当缓存命中（即缓存数据被访问），则将数据移到链表头部
3. 当链表满的时候，将链表尾部的数据丢弃



LFU

Least Frequently Used（最近最经常使用）如果一个数据在最近一段时间很少被访问到，那么可以认为在将来它被访问的可能性也很小。因此，当空间满时，最小频率访问的数据最先被淘汰。



FIFO

FIFO按照“先进先出（First In, First Out）”的原理淘汰数据，正好符合队列的特性，数据结构上使用队列Queue来实现

实现：

1. 新访问的数据插入FIFO队列尾部，数据在FIFO队列中顺序移动；
2. 淘汰FIFO队列头部的数据；

特点：

1. 命中率 命中率很低，因为命中率太低，实际应用中基本上不会采用。
2. 简单 实现代价很小

LRU和LFU的区别

- LRU淘汰时淘汰的是链表最末尾的数据。而LFU是一段时间内访问次数最少的。
- LRU是最近最少使用页面置换算法(Least Recently Used),也就是首先淘汰最长时间未被使用的页面!
- LFU是最近最不常用页面置换算法(Least Frequently Used),也就是淘汰一定时期内被访问次数最少的页!

使用了缓存淘汰策略后，程序不会报错，但是会出现个别用户购物车数据丢失的情况。这时就需要去衡量下，到底是新加购物车数据重要，还是其他已经存在购物车的数据比较重要。可以考虑使用删除最老的数据allkeys-lru策略。当内存满了时，把几年前用户的购物车数据清掉也无伤大雅。

如果一个key放不下了怎么办

还有一种情况是一个用户的购物车数据放不下了。

首先必须要知道为什么会出现这种情况。主要是因为使用String类型值存储数据。而String类型值最大为512M。所以当Key的值到达512M时是无法继续存放的。

所以真的为了防止出现这种情况，在不考虑JVM内存溢出的情况下。可以考虑对于购物车业务中添加数据时增加数据大小判断。如果值到达一个我们设置的上限，例如300M时，重新放入到一个新的Key中。

这个Key在设置时可以设置为 `cart:用户ID:N` 其中N是从1开始的整数。

这样用户购物车的数量就可以无限大了。