redis的过期策略以及内存淘汰机制

redis采用的是定期删除+惰性删除策略。

为什么不用定时删除策略？

定时删除，用一个定时器来负责监视key，过期则自动删除。虽然内存及时释放，但是十分消耗cpu资源。在大并发请求下，CPU要将时间应用在处理请求，而不是删除key，因此没有采用这一策略。

定期删除+惰性删除是如何工作的呢？

定期删除，redis默认每个100ms检查是否有过期的key，有则删除。需要说明的是，redis不是每个100ms将所有的key检查一次，而是随机抽取进行检查（如果每隔100ms检查全部key，redis就会被卡死了）。因此，如果只采用定期删除策略，会导致很多key到时间没有删除。

于是，惰性删除就派上用场了。也就是说在你获取某个key的时候，redis会检查一下，这个key如果设置了过期时间那么是否过期了？如果过期了此时就会删除。

采用定期删除+惰性删除是否就没有其他问题了？

不是的，如果定期删除时没删除key，然后你也没及时去请求key，也就是说惰性删除也没生效，这样，redis的内存会越来越高。那么就应该采用内存淘汰机制。

在redis.conf中有一行配置：

# maxmemory-policy volatile-lru

这个配置就是配内存淘汰策略的：

1. noeviction:

当内存不足以容纳新写入数据时，新写入操作会报错。

1. allkeys-lru:

当内存不足以容纳新写入数据时，在键空间中，移除最少使用的key。推荐使用。

1. allkeys-random:

当内存不足以容纳新写入数据时，在键空间中，随机移除某个key。

1. volatile-lru:

当内存不足以容纳新写入数据时，在设置了过期时间的键空间中，移除最少使用的key。这种情况一般是把redis既当缓存，又做持久化存储的时候才用。不推荐。

1. volatile-random:

当内存不足以容纳新写入数据时，在设置了过期时间的键空间中，随机移除某个key。依然不推荐。

1. volatile-ttl:

当内存不足以容纳新写入数据时，在设置了过期时间的键空间中，有更早过期时间的key优先移除。不推荐。

PS：如果没有设置expire的key，不满足先决条件(prerequisites)；那么volatile-lru、volatile-random和volatile-ttl策略的行为，和noeviction（不删除）基本上一致。