



链滴

面试题之 Redis 篇

作者: [zhengliwei](#)

原文链接: <https://ld246.com/article/1637402566852>

来源网站: 链滴

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

Redis支持哪些数据类型

- string
- list
- set
- hash
- zset (sorted set)

list和set数据类型的区别

- list有序，set无序
- list可以有重复元素，set不能有重复元素

用Redis实现分布式锁的关键点

- 加锁用setnx key thread_id expire 10s，其中：只有key不存在时才加锁，实现锁的基本定义，value中放占用锁的线程ID，用于解锁时校验，同时设置超时时间，防止代码出问题导致锁一直不释放，一般超时时间要远大于正常程序执行时间；
- 解锁用lua脚本，保证解锁操作的原子性，先判断key的value是不是当前线程ID，如果不是，则返回，如果是，才解锁；

zset的底层数据结构

- 当元素少，且zset总大小比较小的时候，底层用的是ziplist，压缩列表，压缩列表占用内存少，但插入和查询元素时只能遍历，复杂度为 $O(n)$ ，在元素少时比较好；
- 当元素超过一个阈值或大小超过一个阈值时，底层会转成skiplist，跳表，跳表占用内存比压缩列表，但插入和查询元素时复杂度为 $O(\log n)$ ，性能更好，且相比于红黑树，跳表实现更简单，这也是redis选择跳表的原因之一。

Redis过期键的清理策略

- Redis有两种清理策略：惰性清理和定期清理；
- 惰性清理是指在查询元素时先检查元素是否过期，如果过期了，就删除元素并返回不存在；
- 定期清理是指后台定期启动子线程去执行清理过期键的逻辑，并且清理时会采用随机抽取元素，检查是否过期，如果过期，则清理，同时，检查这次抽取的元素过期比例，如果比例大于某个阈值，则继续清理过程，如果小于阈值，则停止清理。

为什么要使用两种清理策略

- 惰性清理对cpu友好，对内存不友好，如果只使用惰性清理，可能会浪费内存；
- 定期清理对内存友好，对cpu不友好；如果只使用定期清理，可能会导致清理时redis反应慢，甚至死；
- 两种策略结合，可以在保证清理逻辑不影响业务使用的同时，避免过期键占用过多内存。