压缩列表

01 / 压缩列表简介

1. 压缩列表（ziplist），是Redis为了节约内存而设计的一种线性数据结构，它是由一系列具有特殊编码的连续内存块构成的；
2. 一个压缩列表可以包含任意多个节点，每个节点可以保存一个字节数组或一个整数值。

Redis的列表、哈希、有序集合都直接或间接地使用了压缩列表！

- 压缩列表的结构示意：

- 压缩列表的组成说明：

属性	类型	长度	说明
zlbytes	uint32_t	4字节	压缩列表占用的内存字节数；
zltail	uint32_t	4字节	压缩列表表尾节点距离列表起始地址的偏移量（单位字节）；
zllen	uint16_t	2字节	压缩列表包含的节点数量，等于UINT16_MAX时，需遍历列表计算真实数量；
entryX	列表节点	不固定	压缩列表包含的节点，节点的长度由节点所保存的内容决定；
zlend	uint8_t	1字节	压缩列表的结尾标识，是一个固定值0xFF；

- 压缩列表的节点构成：

previous_entry_length	encoding	content

- previous_entry_length

  该属性以字节为单位，记录当前节点的前一节点的长度，其自身占据1字节或5字节：

  1. 如果前一节点的长度小于254字节，则“pel”属性的长度为1字节，前一节点的长度就保存在这一个字节内；
  2. 如果前一节点的长度达到254字节，则“pel”属性的长度为5字节，其中第一个字节被设置为0xFE，之后的四个字节用来保存前一节点的长度；

  基于“pel”属性，程序便可以通过指针运算，根据当前节点的起始地址计算出前一节点的起始地址，从而实现从表尾向表头的遍历操作。

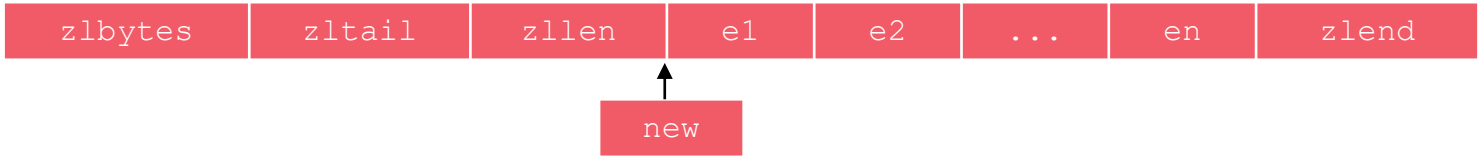
- encoding & content

  content属性负责保存节点的值（字节数组或整数），其类型和长度则由encoding属性决定。

encoding	长度	content
00 xxxxxx	1字节	最大长度为2^6 -1的字节数组;
01 xxxxxx bbbbbbbb	2字节	最大长度为2^{14}-1的字节数组;
10 ______ bbbbbbbb	5字节	最大长度为2^{32}-1的字节数组;
11 000000	1字节	int16_t类型的整数;
11 010000	1字节	int32_t类型的整数;
11 100000	1字节	int64_t类型的整数;
11 110000	1字节	24位有符号整数;
11 111110	1字节	8位有符号整数;
11 11xxxx	1字节	没有content属性，xxxx直接存[0,12]范围的整数值;

- 添加引起的连锁更新

  1. e1~en节点的长度均介于250字节~253字节之间;
  2. 将一个长度大于等于254字节的节点new添加至表头;



- 删除引起的连锁更新

  1. e1~en节点的长度均介于250字节~253字节之间;
  2. big节点长度大于等于254，small节点长度小于254，将small节点删除;

- 连锁更新的影响

1. 在最坏情况下，连锁更新需要对压缩列表执行N次空间重分配操作；
2. 每次空间重分配的最坏复杂度为O(N)，所以连锁更新的最坏复杂度为O(N^2)。

1. 连锁更新出现的概率很低：

   压缩列表中需要恰好有多个连续的，长度介于250~253字节的节点；

2. 控制节点数量可消除影响：

   如果节点数量不多，即便出现连锁更新，对性能也不会造成任何影响。

牛客大学
-专业求职辅导-

THANKS

关注【牛客大学】公众号
回复“牛客大学”获取更多求职资料