

Redis数据类型

String类型

特征：即存储字符串的数据类型，单个字符串存储最大不超过512MB

常使用业务场景：用来存储JSON序列化之后对象

底层编码：

Int编码

数据结构特点：ptr指针直接指向字符串常量池中对应字符串地址，而且数组中元素只能为数字

使用要求：每一个字符必须是数字

EmbStr编码

数据结构特点：采用简单动态字符串结构，这种结构的特征就在于是一个可扩容的字符数组，由当前已使用长度字段，当前未使用长度字段以及一个字符数组组成。

扩容规则：当字符数组大小小于1MB时，将总长度进行翻倍，如果字符数组大小大于1MB，那么每次增加1MB的长度

使用要求：当字符串大小小于32MB时候，采用的编码

数据结构内存地址所在：为RedisObject对象后的连续内存

Raw编码

数据结构特点：同样采用简单动态字符串作为数据结构

使用要求为当字符串大小超过32MB的时候，自动转为RAW编码

数据结构内存地址所在：不和RedisObject对象在一起，而是在内存中其他随机地方

重点：由于EmbStr和Raw编码中简单动态字符串结构(SDS)内存地址不同，所以最开始生成String对象的时候，分配内存的次数也不同，由于EmbStr的RedisObject对象和SDS对象是连续的，所以一起分配内存就行，但是raw需要两次内存分配

List类型

特征：为有序，可重复列表。每个List类型可以插入(2的32次方)-1个元素

常用业务场景

最新与最老数据的TOP显示

由于底层采用双向链表/压缩列表的方式进行存储，均能获取头节点和尾节点指针，每个节点均有前后指针的关系，非常适合进行操作头部与尾部节点

消息队列

即生产者可以使用Lpush方法往List类型的头部插入信息，然后消费者通过Lpop对List类型的尾部进行消费

底层编码

ZipList编码

数据结构特点：即一段连续的内存数组，但是和普通的数组不同，他的每一个元素空间都是根据其中存储的元素大小来规定的，并不是死板的固定大小，并有记录当前最后一个元素位置的指针。

使用要求：即当每一个元素大小不超过64个字节，总元素个数不超过512个

数据结构内存地址所在：作为RedisObject对象后的连续内存，所以生成List对象时，此编码只用分配一次内存

LinkedList编码

数据结构特点：就是双向链表

使用要求：当不满足zipList编码要求的时候，就会变成LinkedList编码

数据结构内存地址所在：不和RedisObject在一起，而是在内存中的任意位置

Hash类型

特征：每个Hash类型可以存放(2的32次方)-1个键值对

常用业务场景：

购物车的管理

即使用用户id作为key值，用商品id作为field字段，用商品数量作为value，那么我们可以使用hgetAll指令完成购物车全选，hdel删除指定商品等等操作

底层编码

zipList编码

数据结构特点：和List类型采用此编码时运用不太一样，Hash类型会把field-value键值对连续的插入到压缩列表中，从而完成不同键值对的存储

使用要求：即当每一个元素大小不超过64个字节，总元素个数不超过512个

数据结构内存地址所在：作为RedisObject对象后的连续内存，所以生成List对象时，此编码只用分配一次内存

Hashtable编码

数据结构特点：底层为哈希表数据结构，用以存储field-value键值对

使用要求：当不满足zipList编码要求的时候，就会变成LinkedList编码

Set类型

特征：为无序，而不可重复集合，可以存放(2的32次方)-1个元素

常使用业务场景：

元素去重

即大量元素去重操作会占用大量CPU资源，那么把这些数据存储到Set类型中，即可完成去重操作

共同好友等

把一个用户的好友和另一个用户的好友进行交集处理，得到他们的共同好友，然后进行推荐

底层编码

inset编码

数据结构特点：即由整数数组组成的连续内存(此数组按元素大小进行排序)，但是除了整数数组外，还有有当前数组包含的元素数量以及采用整形的编码，可分为int16，int32，int64.分别是整数数组中运行存放数值的大小

使用要求：元素个数小于512个，而且集合内元素全为数字

数据结构内存地址所在：和RedisObject相连的连续内存

HT编码

数据结构特点：由Value值为空的Hashtable结构组成，被称为字典。

使用要求：当不满足inset编码要求的时候，就会变成HT编码

ZSet类型

特征：为有序，而不可重复集合，可以存放(2的32次方)-1个元素

常使用业务场景：

热门数据TOP排行榜

即利用Zset类型，有序不重复特点，把重复的数据进行去重并且使用score字段排序后，从小到大排序，反序即可得到热门数据TOPC排行榜

底层编码

zipList编码

数据结构特点：压缩列表的另一种使用，即使用一个新增的score字段来决定元素在压缩列表中的位置，默认从小到大

使用要求：元素单个大小不能超过64字节，压缩数组长度不能超过128个元素

数据结构内存地址所在：和RedisObject相连的连续内存

skipList编码

数据结构特点：通过跳表与Hashtable的集合编码，利用跳表存储hash表中的节点来完成有序，操作主要还是在hash表中，跳表中的大小由Score字段来决定

使用要求：当不满足zipList编码要求的时候，就会变成skipList编码