Redis管理

数据结构

KeyserCache号外1.5发!

不使用C语言的字符串（动态字符串），并提高redis安全性和性能

char *buf = NULL;
sdsnewlen("redis", 5);
sdscat(buf, " server");
printf("%s\n", buf);
sdsfree(buf);

- 字符串长度：O(1)
- 二进制安全
- 可以存储一些二进制数据，例如序列化的对象
- 字符串最大长度为512M

- 数据结构
  - linked list（双向无环带表头表尾指针的链表）
  - 带有长度计数器
  - 链表节点保存的值可以是一个字符串、一个整数值、一个双精度浮点数、一个指向其他数据结构的指针
  - 链表可以用于保存各种不同类型的值，也可以保存各种不同类型的数据
  - 链表的长度最大为2^32-1个节点
  - 链表提供了获取表头、表尾、某个节点的前置节点、后置节点、给定值的节点的O(1)复杂度操作
  - 链表是一个双端、无环、带表头节点和表尾指针、带长度计数器的双向链表
  - 链表的每个节点由一个listNode结构来表示
  - 每个listNode结构的prev和next指针分别指向该节点的前置节点和后置节点
  - 每个listNode结构的value属性用于保存节点的值
  - 链表的长度由list结构的length属性记录
  - 链表的表头指针由list结构的head指针记录，表尾指针由list结构的tail指针记录
  - 链表的dup、free、match三个属性分别用于为链表设置复制函数、释放函数、比较函数
  - 字典（dict）
  - 字典的实现使用了两个哈希表（ht[0]、ht[1]）
  - 字典使用ht[0]哈希表来进行日常工作，而ht[1]则在进行rehash时使用
  - 字典的rehash是通过将ht[0]中的键值对rehash到ht[1]中来实现的
  - 字典的rehash是分多次、渐进地进行的
  - 字典的rehash会在服务器空闲时候执行
  - 字典的rehash会在ht[0]的负载因子大于5时，或者服务器没有执行BGSAVE命令或BGREWRITEAOF命令的情况下，开始执行扩展操作
  - 字典的rehash会在ht[0]的负载因子小于0.1时，开始执行收缩操作
  - 字典的键值对的删除、查找、更新等操作，都会在指向较小哈希表的ht[0]和ht[1]两个哈希表上进行
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，会同时对两个哈希表进行操作
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，还可能会对一个哈希表进行操作
  - 字典的键值对的删除、查找、更新等操作，在字典没有进行rehash的期间，只会对一个哈希表进行操作
  - 字典的键值对的删除、查找、更新等操作，在字典没有进行rehash的期间，平均时间复杂度为O(1)
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，平均时间复杂度仍然为O(1)
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，最坏时间复杂度为O(N)
  - 字典的键值对的删除、查找、更新等操作，在字典没有进行rehash的期间，最坏时间复杂度为O(2)
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，最坏时间复杂度为O(3)
  - 字典的键值对的删除、查找、更新等操作，在字典没有进行rehash的期间，最坏时间复杂度为O(1)
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，最坏时间复杂度为O(2)
  - 字典的键值对的删除、查找、更新等操作，在字典正在进行rehash的期间，最坏时间复杂度为O(3)
  - 字典的键值对的删除、查找、更新等操作，在字典正在