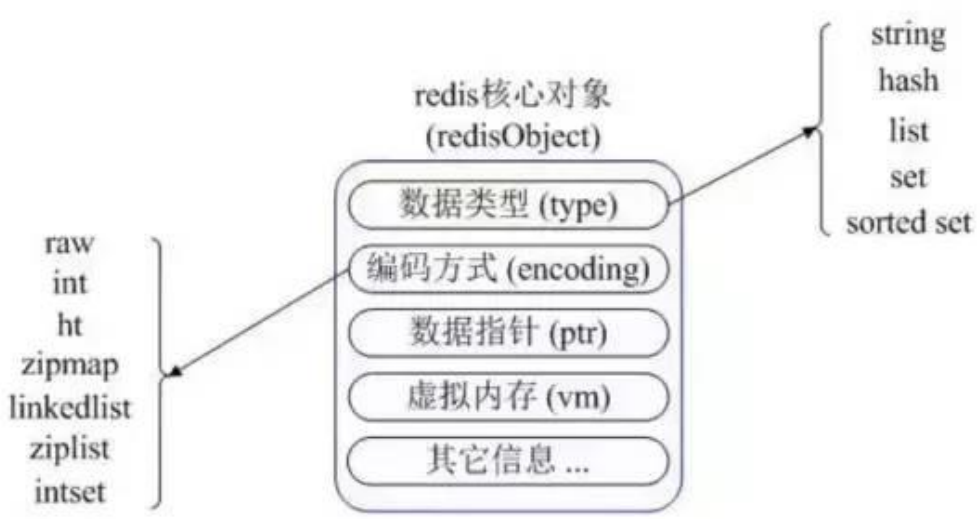
什么是Redis？

> Redis 是一个使用 C 语言写成的，开源的 **key-value** 数据库。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。

Redis内部内存管理：



- Redis内部使用一个`redisObject`对象来表示所有的key和value。
  - type：一个value对象是何种数据类型
  - encoding：不同数据类型在redis内部的存储方式

5种数据类型：

1. **String**是redis最基本的类型，最大存储512M

   常用命令: *set,get,decr,incr,mget* 等。

   使用场景: 常规计数：微博数，粉丝数等.二进制安全，可以存储任何对象，比如jpg图片或者序列化对象。

2. **Hash**是一个键值的集合，适合存储对象

   常用命令： *hget,hset,hgetall* 等

   使用场景：存储、读取、修改用户属性(电商商品缓存)

3. **List**简单的字符串列表，按照插入顺序排序

   常用命令: *lpush,rpush,lpop,rpop,lrange*等

   实现：双向链表，支持反向查找和遍历，方便操作，但是带来了额外的内存开销。

   常用命令: lpush,rpush,lpop,rpop,lrange等

   使用场景：最新消息排行；消息队列，某某列表

4. **Set**：String类型的无序集合

常用命令： *sadd,spop,smembers,sunion* 等

实现：**hashtable**实现的，添加、删除、查找的复杂度都是O(1)，提供了求交集、并集、差集的操作

共同好友，利用唯一性统计访问网站的所有ip

5. **Sorted Set**：通过用户额外提供一个优先级score的double类型参数来为成员排序，并且是插入即有序，自动排序的。

   常用命令： zadd,zrange,zrem,zcard等

   实现：hashmap+skiplist跳跃表，hashmap中存放成员到score的映射，skiplist中存放所有成员，排序依据是hashmap中的score，使用跳跃表可以获得较高的查找效率，实现上简单。

   使用场景：带权重的消息队列，排行榜

类型	简介	特性	场景
String(字符串)	二进制安全	可以包含任何数据,比如jpg图片或者序列化的对象,一个键最大能存储512M	---
Hash(字典)	键值对集合,即编程语言中的Map类型	适合存储对象,并且可以像数据库中update一个属性一样只修改某一项属性值(Memcached中需要取出整个字符串反序列化成对象修改完再序列化存回去)	存储、读取、修改用户属性
List(列表)	链表(双向链表)	增删快,提供了操作某一段元素的API	1,最新消息排行等功能(比如朋友圈的时间线) 2,消息队列
Set(集合)	哈希表实现,元素不重复	1、添加、删除,查找的复杂度都是O(1) 2、为集合提供了求交集、并集、差集等操作	1、共同好友 2、利用唯一性,统计访问网站的所有独立ip 3、好友推荐时,根据tag求交集,大于某个阈值就可以推荐
Sorted Set(有序集合)	将Set中的元素增加一个权重参数score,元素按score有序排列	数据插入集合时,已经进行天然排序	1、排行榜 2、带权重的消息队列

使用redis有哪些好处？

1. **速度快**，因为数据存在内存中，类似于HashMap，HashMap的优势就是**查找和操作的时间复杂度都是O(1)**
2. **支持丰富数据类型**，支持string，list，set，sorted set，hash
3. **支持事务**：redis对事务是部分支持的(不支持回滚)，如果是在入队时报错，那么都不会执行；在非入队时报错，那么成功的就会成功执行。
4. redis监控：锁的介绍