

1.五中基本数据结构

- String : 字符串
- 字段Hash
- 列表List
- 集合Set
- 有序集合SortedSet

2.高级数据结构

2.1 HyperLogLog

Redis HyperLogLog 是用来做基数统计的算法，HyperLogLog 的优点是，在输入元素的数量或者体积非常非常大时，计算基数所需的空间总是固定的、并且是很小的。

什么是基数：比如数据集 {1, 3, 5, 7, 5, 7, 8}，那么这个数据集的基数集为 {1, 3, 5, 7, 8}，基数(不重复元素)为5。基数估计就是在误差可接受的范围内，快速计算基数。

```
*****
[root@redisuav-fb90a123 ~]# redis-cli
127.0.0.1:6379> pfadd runoobkey "redis"
(integer) 1
127.0.0.1:6379> pfadd runoobkey "mongodb"
(integer) 1
127.0.0.1:6379> pfadd runoobkey "mysql"
(integer) 1
127.0.0.1:6379> pfcount runoobkey
(integer) 3
127.0.0.1:6379> █
```

适合什么场合用

记录网站每天访问的独立IP数量这样的一个功能

HyperLogLog 除了上面的 pfadd 和 pfcount 之外，还提供了第三个指令 pfmerge，用于将多个 pf 计数值累加在一起形成一个新的 pf 值。

比如在网站中我们有两个内容差不多的页面，运营说需要这两个页面的数据进行合并。其中页面的 UV 访问量也需要合并，那这个时候 pfmerge 就可以派上用场了。

2.2 Geo

可以将用户给定的地理位置信息储存起来，并对这些信息进行操作

目前 redis 支持以下 6 个 GEO 的相关操作：

- geoaddd：增加地理位置的时候，会先计算地理位置坐标的geohash值，然后地理位置作为有序集合的member，geohash作为该member的score。然后使用zadd命令插入到有序集合。
- geopos：先根据地理位置获取geohash值，然后decode得到地理位置的坐标。
- geodist：先根据两个地理位置各自得到坐标，然后计算两个坐标的距离。
- georadius和georadiusbymember使用相同的实现，georadiusbymember多了一步把地理位置转换成对应的坐标。然后查找该坐标和周围对应8个坐标符合距离要求的地理位置。因为geohash得到的值其实是个格子，并不是点，这样通过计算周围对应8个坐标就能解决边缘问题。由于使用有序集合保存地理位置，在对地理位置基于范围查询，就相当于实现了zrange命令，内部的实现确实与zrange命令一致，只是geo有些特别的处理，比如获得的某个地理位置，还需要计算该地理位置是否符合给定的距离访问。
- geohash：获取某个地理位置的geohash值。

场景：无人机飞行距离计算

3. 高版本支持数据结构

3.1.1 BloomFilter

自Redis 4.0以来，ReBloom模块已经可用，它可以消除任何Bloom过滤器实现开销

基于redis的Bloomfilter去重，既用上了Bloomfilter的海量去重能力，又用上了Redis的可持久化能力，基于Redis也方便分布式机器的去重。在使用的过程中，要预算好待去重的数据量，则根据上面的表，适当地调整seed的数量和blockNum数量（seed越少肯定去重速度越快，但漏失率越大）。

3.1.2 RedisSearch

自Redis 4.0以来支持RedisSearch。

Redisearch在Redis上面实现了一个搜索引擎，但与其他Redis搜索库不同，它不使用内部数据结构，如排序集。这也可以实现更高级的功能，如文本查询的完全词组匹配和数字过滤，这对传统的redis搜索几乎是不可能或高效的。

3.1.3 JSON

ReJSON 是一个Redis Module，它实现了ECMA-404 The JSON Data Interchange Standard作为本地数据类型，它允许从Redis Keys（documents）中存储，更新和获取JSON值

主要特性：

- 完全支持JSON标准
- 对于在文档内选择元素类似JSONPath语法
- 文档作为二进制数据被存储在一个树形结构中，允许快速访问子元素
- 对所有JSON数据类型按照原子操作进行分类