Redis介绍

xingxing.wang

What Redis ?

1. 开源免费, ANSI C实现

2. Key-value 类型数据的分布式NoSQL数据库系统

3. 高性能，持久存储，适应高并发的应用场景

4. 优点：

   a. 性能极高 – 支持超过 100K+ 每秒的读写频率

   b. 丰富的数据类型 – 支持二进制案例的 Strings, Lists, Hashes, Sets 及 Ordered Sets 数据类型操作

   c. 原子 – 所有操作都是原子性的，同时Redis还支持对几个操作全并后的原子性执行

   d. 丰富的特性 – 支持 publish/subscribe, 通知, key 过期等等特性

数据结构

1. String 字符串类型

   a. SET, GET命令

   文档：http://redis.readthedocs.org/en/latest/index.html

   b. value为整形数字时：INCR, INCRBY, DECR, DECRBY

   c. Redis key是String，二进制安全的，可以用任何二进制序列作为key值[JPEG文件都可以]

   d. 常用对象共用

   redis server启动时初始化10000个Integer字符串对象[0,9999]

数据结构

2. List类型

a. 基于Linked Lists实现

b. LPUSH, RPUSH

c. LINDEX, LRANGE

3. Set集合

a. 未排序的集合，不可重复

b. SADD, SCARD, SDIFF, SPOP

c. SORT命令可使用于 Set 和 List

数据结构

4. Sorted Set类型

   a. score属性，字符型，写入时就按这个score排好序

   b. ZADD, ZCARD, ZRANGE

5. Hash类型

   a. 存储key对多个属性的数据

   b. HSET, HGET, HKEYS, HDEL, HMSET, HMGET

内存数据结构

1. 简单动态字符串 （simple dynamic string，SDS）

a. 传统C字符串以'\0'结束，无边界检查

b. SDS记录自身长度，len()操作时间复杂度O(1)，可以便捷检查

c. 空间预分配

1）char* buf 字符缓存

2）free 未使用字节数量

3）分配的字节数大小根据需求计算，并不等于申请的实际字节数，一般大于等于实际申请字节数

d. 惰性空间释放

缩短 SDS 保存的字符串时， 程序并不立即使用内存重分配来回收缩短后多出来的字节， 而是使用 free 属性将这些字节的数量记录起来，并等待将来使用

内存数据结构

2. 链表
   a. 双端链表

   b. 链表用list 结构来表示， 这个结构带有表头节点指针、表尾节点指针、以及链表长度等信息

   c. 无环链表
      表头节点的前置节点和表尾节点的后置节点都指向 NULL

3. 字典
   a. 使用哈希表作为底层实现

   b. 链地址法来解决键冲突:
      被分配到同一个索引上的多个键值对会连接成一个单向链表

   c. rehash: 维持哈希表的负载因子合理
      两个哈希表， 一个用于平时使用， 另一个仅在进行 rehash 时使用

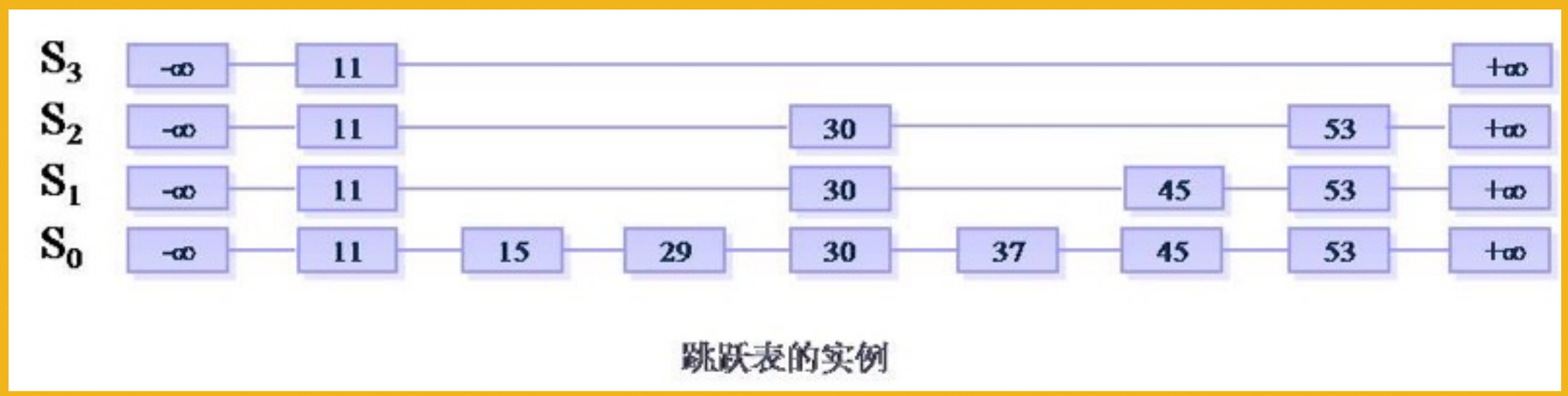
内存数据结构

4. 跳跃表

a. 可替代平衡树，有序集合的底层实现

b. 节点按照分值大小进行排序， 当分值相同时， 节点按照成员对象的大小进行排序

c. http://blog.sina.com.cn/s/blog_60707c0f0100wudj.html



跳跃表的实例

内存数据结构

5. 整数集合

a. 集合键的底层实现之一

b. 底层实现为数组， 这个数组以有序、无重复的方式保存集合元素

6. 压缩列表

a. 为节约内存而开发的顺序型数据结构

b. 列表键和哈希键的底层实现之一

c. 可以包含多个节点，每个节点可以保存一个字节数组或者整数值

内存数据结构

7. 对象类型

a. 每个键值对的键和值都是一个对象

b. 字符串、列表、哈希、集合、有序集合五种类型的对象，每种对象拥有不同的编码方式，不同的编码可以在不同的使用场景上优化对象的使用效率

c. 引用计数实现的内存回收机制, 当一个对象不再被使用时，该对象所占用的内存就会被自动释放

Publish/Subscribe

将数据推到某个信息管道中，然后其它人可以通过订阅这些管道来获取推送过来的信息

http://redis.readthedocs.org/en/latest/pub_sub/index.html

数据过期设置

1. EX second： 设置键的过期时间为 second 秒
2. PX millisecond： 设置键的过期时间为 millisecond 毫秒
3. NX： 只在键不存在时，才对键进行设置操作
4. XX： 只在键已经存在时，才对键进行设置操作

事务性

1. 文档：http://redis.readthedocs.org/en/latest/topic/transaction.html

2. MULTI, EXEC, DISCARD, WATCH

3. 有命令在入队时失败，那么大部分客户端都会停止并取消这个事务

4. 即使事务中有某条/某些命令执行失败了， 事务队列中的其他命令仍然会继续执行 —— Redis 不会停止执行事务中的命令

复制

1. 文档：http://redis.readthedocs.org/en/latest/topic/replication.html

2. slaveof： 读写分离

持久化

1. 文档:
   a. http://redis.readthedocs.org/en/latest/topic/persistence.html

   b. http://www.cnblogs.com/wenanry/archive/2012/02/26/2368398.html

2. RDB: 在指定的时间间隔内生成数据集的时间点快照

   a. save: 多长时间，多少次更新操作，将数据同步到数据文件

   b. rdbcompression yes/no

3. AOF: 记录服务器执行的所有写操作命令，并在服务器启动时，通过重新执行这些命令来还原数据集

   a. appendonly yes/no

   b. appendfsync no/always/everysec 官方推荐每秒一次：everysec

Jedis

1. Redis Java Client

2. 对象池：GenericObjectPool

3. value：String，byte[]
   a. 序列化
   b. Json

简单的锁

1. 获取锁：

SET resource-name anystring NX EX max-lock-time

2. 释放锁：

不直接DEL key，传入的值和键的口令串相匹配时，才对键进行删除