Redis 笔记

郑华

2018 年 11 月 24 日

第一章　Redis 简介

1.1　参考文献

http://www.runoob.com/redis/redis-backup.html
进阶：https://www.cnblogs.com/hjwublog/p/5660578.html

1.2　应用场景

- 缓存
- 聊天室，秒杀，任务队列
- 应用排行榜. 网站统计
- 数据存储 (add,del,update,select) 定期持久化到硬盘中
- 分布式集群架构中的 session 分离

1.3　安装与启动

```
// 安装
$ wget http://download.redis.io/releases/redis-2.8.17.tar.gz
$ tar xzf redis-2.8.17.tar.gz
$ cd redis-2.8.17
$ make


//启动redis服务.
$ cd src
$ ./redis-server

//使用默认配置启动redis 服务
./redis-server redis.conf

//启动redis服务进程后，就可以使用测试客户端程序redis-cli和redis服务交互了
./redis-cli
```

第二章　基本操作

redis 命令**不区分大小写**，所以`get var` 和`GET var` 是等价的

2.1　选库

使用 **Select** 命令用于*切换到***指定的数据库**，数据库索引号 index 用数字值指定，以 0 作为起始索引值。

2.2　Key

表 2.1: 常用命令

命令	含义
keys [pattern]	返回相应的的 key
set key value	设定一个 key-value
get key	获取一个 key 的值
dump key	序列化给定 key, 并返回序列化的值
randomkey	返回随机的 key
exists key	判断 key 是否存在
type key	返回 key 存储的类型
del key	删除 key
expire key seconds	给 key 设置过期时间
persist key	移除 key 的过期时间，key 将持久保存

第三章　数据类型

3.1　string

二进制安全的。意思是 redis 的 string 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512MB

- `set key value [ex 秒数]|[px 毫秒数]  [nx]|[xx]` : 设置键值
  - `nx` 表示 key 不存在时执行操作
  - `xx` 表示 key 存在时执行操作
  - `ex` 表示设置过期时间
  - `px` 表示设置持续时间，`ex` 与`px` 不能同时设置
- `mset key1 v1 key2 v2 ...` : 一次性设置多个键值对 (multi set)
- `get key` 获取 key 的值
- `mget key1 key2 ...` : 一次性获取多个键的值
- `setrange key offset value` : 把字符串的`offset` 偏移字节改为`value`

```
set greet hello
get greet --> hello
setrange greet 2 x
get greet --> hexlo

setrange greet 2 ??
get greet --> he??o

// 当偏移量大于字符长度时，多余部分将自动以0x00 填充
setrange greet 6 !
get greet --> he??o0x00!
```

- `append key value` : 把 value 追加到 key 的原值上
- `getrange key start stop` : 获取字符串中 `start, stop` 范围的值
  - 左闭右闭区间，从 0 开始

– 负数表示倒数第多少

- `getset key newValue`：获取并返回旧值，并设置新值
- `incr|decr key`：增 1 或减 1,不存在的 key 当成 0 再 incr 返回。与之对应的有`incrby key num`
- `getbit key offset`：获取值的二进制的对应位上的值，从高位开始
- `setbit key offset value`：设置对应 2 进制位上的值，offset 最长能达到`2^32-1` 位
- `bitop op destKey key1 [key2 key3 ...]`：对 [key1 key2..] 做 op，并将结果保存到 destKey 中，op 可以为一下几种`AND OR NOT XOR`

3.2 list

按照插入顺序排序。你可以添加一个元素到列表的头部（左边）或者尾部（右边）

表 3.1: list 操作说明

命令	含义
lpush key value1[value2]	
lpop key	
blpop key1[key2] timeout	
brpop key1[key2] timeout	
lindex key index	
linsert	
llen key	
lpushx key value	
lrange key start stop	
lset key index value	
ltrim key start stop	
rpop key	
rpoplpush	
rpush key value1[value2]	
rpushx key value	

3.3 hash

表 3.2: hash 操作说明

命令	含义
hdel key field[field2]	
hexists key filed	
hget key field	
hincrby key field increment	
hkeys key	
hlen key	
hmget key field1[filed2]	
hmset key field1[field2]	
hset key field value	
hsetnx key field value	
hvals key	

3.4 set

Set 是 String 类型的无序集合。集合成员是唯一的，这就意味着集合中不能出现重复的数据。Redis 中集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是 O(1)。

表 3.3: set 操作说明

命令	含义
SADD key mem1 [m2]	
SCARD key	
SDIFF key1 [key2]	
SINTER key1 [key2]	
SISMEMBER key mem	
SMEMBERS key	
SMOVE sr des mem	
SPOP key	
SRANDMEMBER key	
SREM key mem1 [mem2]	
SUNION key1 [key2]	

3.5 zset

有序集合和集合一样也是 string 类型元素的集合, 且不允许重复的成员。

不同的是每个元素都会关联一个 double 类型的分数。redis 正是通过分数来为集合中的成员进行从小到大的排序。

有序集合的成员是唯一的, 但分数 (score) 却可以重复。

集合是通过哈希表实现的，所以添加，删除，查找的复杂度都是 O(1)。

表 3.4: zset 操作说明

命令	含义
ZADD	
ZCARD	
ZCOUNT	
ZINCRBY	
ZRANGE	
ZRANGEBYLEX	
ZRANGEBYSCORE	
ZRANK	
ZREM	
ZREMRANGEBYLEX	
ZREVRANK	

3.6 事务与锁

类似于 mysql 的 start transaction，可以保证原子性，可以使用 rollback 取消操作。

redis 使用 multi 命令实现

使用 watch 锁，解决多用户竞争

Redis 事务可以一次执行多个命令，并且带有以下两个重要的保证：

- 批量操作在发送 EXEC 命令前被放入队列缓存。
- 收到 EXEC 命令后进入事务执行，事务中任意命令执行失败，其余的命令依然被执行。
- 在事务执行过程，其他客户端提交的命令请求不会插入到事务执行命令序列中。

一个事务从开始到执行会经历以下三个阶段：

1. 开始事务。
2. 命令入队。
3. 执行事务。

表 3.5: 事务与锁操作说明

命令	含义
MULTI	标记一个事务块的开始。
EXEC	执行所有事务块内的命令。
DISCARD	取消事务，放弃执行事务块内的所有命令。
UNWATCH	取消 WATCH 命令对所有 key 的监视。
WATCH key [key ...]	监视一个 (或多个) key ，如果在事务执行之前这个 (或这些) key 被其他命令所改动，那么事务将被打断。

第四章　订阅与发布

第五章　持久化

5.1　rdb 持久化

SAVE|BGSAVE　用于创建当前数据库的备份。

恢复数据　如果需要恢复数据，只需将备份文件 (dump.rdb) **移动到 `redis` 安装目录**并启动服务即可。

获取 redis 目录可以使用 CONFIG 命令:`CONFIG GET dir`

5.2　aof 持久化

第六章　分布式

6.1　主从复制

6.2　分布式集群

https://www.cnblogs.com/yuanermen/p/5717885.html

https://www.cnblogs.com/cjsblog/p/9048545.html

https://www.cnblogs.com/liyasong/p/redis_jiqun.html?utm_source=itdadao&utm_medium=referral

6.3　Redis 分区

分区是分割数据到多个 Redis 实例的处理过程，因此每个实例只保存 key 的一个子集。

分区的优势

- 通过利用多台计算机内存的和值，允许我们构造更大的数据库。
- 通过多核和多台计算机，允许我们扩展计算能力；通过多台计算机和网络适配器，允许我们扩展网络带宽。

redis 集群大多数支持在**运行时**增加、删除节点的透明数据平衡的能力，但是类似于客户端分区、代理等其他系统则不支持这项特性。然而，一种叫做presharding 的技术对此是有帮助的。

分区类型

- 范围分区：0-1000,1001-2000，..
- 哈希分区

第七章　应用

7.1　位图法统计活跃用户

- 1 亿个用户

- 如何记录用户的登录信息

- 如何查询活跃用户，一周连续登录

从信息的角度看，一个用户的登录只需要一个位就可以表示。0-1

在数据库中，数据一般都有编号

简化如：

```
7个用户

周一 01011100
周二 01010010
周三 01111000

setbit mon 100000000 0 // 初始化为0
setbit mon 3 1 // 第3号用户登录，标记为1
setbit mon 9 1


setbit tuseday 10000000 0
set bit tuesday 4 1


bitop and mon tuseday ...
```

第八章　PHP 与 Redis

8.1　关联

```php
$redis = new redis();
$result = $redis->connect('127.0.0.1', 6379);
var_dump($result); //结果: bool(true)

$result = $redis->set('test',"11111111111");
var_dump($result); //结果: bool(true)

$result = $redis->get('test');
var_dump($result); //结果: string(11) "11111111111"

$redis->delete('test');
var_dump($redis->get('test')); //结果: bool(false)
```