

Redis-day01-note

王伟超

wangweichao@tedu.cn

Redis介绍

- 特点及优点

- 1 1、开源的，使用C编写，基于内存且支持持久化
- 2 2、高性能的Key-Value的NoSQL数据库
- 3 3、支持数据类型丰富，字符串strings，散列hashes，列表lists，集合sets，有序集合sorted sets 等等
- 4 4、支持多种编程语言（C C++ Python Java PHP ...）

- 与其他数据库对比

- 1 1、MySQL：关系型数据库，表格，基于磁盘，慢
- 2 2、MongoDB：键值对文档型数据库，值为JSON文档，基于磁盘，慢，存储数据类型单一
- 3 3、Redis的诞生是为了解决什么问题？
- 4 # 解决硬盘IO带来的性能瓶颈

- 应用场景

- 1 1、使用Redis来缓存一些经常被用到、或者需要耗费大量资源的内容，通过这些内容放到redis里面，程序可以快速读取这些内容
- 2 2、一个网站，如果某个页面经常会被访问到，或者创建页面时消耗的资源比较多，比如需要多次访问数据库、生成时间比较长等，我们可以使用redis将这个页面缓存起来，减轻网站负担，降低网站的延迟，比如说网站首页等
- 3 # redis的诞生是为了解决负载问题

- redis版本

- 1 1、最新版本：5.0
- 2 2、常用版本：2.4、2.6、2.8、3.0(里程碑)、3.2、3.4、4.0(教学环境版本)、5.0
- 3 3、图形界面管理工具（# 写的一般）
- 4 RedisDesktopManager

- Redis附加功能

```
1 1、持久化
2   将内存中数据保存到磁盘中，保证数据安全，方便进行数据备份和恢复
3 2、过期键功能
4   为键设置一个过期时间，让它在指定时间内自动删除
5   <节省内存空间>
6   # 音乐播放器，日播放排名，过期自动删除
7 3、事务功能
8   原子的执行多个操作
9 4、主从复制
10 5、Sentinel哨兵
```

安装

- Ubuntu

```
1 # 安装
2 sudo apt-get install redis-server
3 # 服务端启动
4 sudo /etc/init.d/redis-server status | start | stop | restart
5 # 客户端连接
6 redis-cli -h IP地址 -p 6379 -a 密码
```

- Windows

```
1 1、下载安装包
2   https://github.com/ServiceStack/redis-windows/blob/master/downloads/redis-
3   64.3.0.503.zip
4 2、解压
5 3、启动服务端
6   双击解压后的 redis-server.exe
7 4、客户端连接
8   双击解压后的 redis-cli.exe
9
10 # windows下产生的问题：关闭终端后服务终止
11 # 解决方案：将Redis服务安装到本地服务
12 1、重命名 redis.windows.conf 为 redis.conf,作为redis服务的配置文件
13 2、cmd命令行，进入到redis-server.exe所在目录
14 3、执行：redis-server --service-install redis.conf --loglevel verbose
15 4、计算机-管理-服务-Redis-启动
16
17 # 卸载
18 到 redis-server.exe 所在路径执行：
19 1、redis-server --service-uninstall
20 2、sc delete Redis
```

配置文件详解

- 配置文件所在路径

```
1 1、Ubuntu
2 /etc/redis/redis.conf
3 mysql的配置文件在哪里? : /etc/mysql/mysql.conf.d/mysqld.cnf
4
5 2、windows 下载解压后的redis文件夹中
6 redis.windows.conf
7 redis.conf
```

- 设置连接密码

```
1 1、requirepass 密码
2 2、重启服务
3 sudo /etc/init.d/redis-server restart
4 3、客户端连接
5 redis-cli -h 127.0.0.1 -p 6379 -a 123456
6 127.0.0.1:6379>ping
```

- 允许远程连接

```
1 1、注释掉本地IP地址绑定
2 69行: # bind 127.0.0.1 ::1
3 2、关闭保护模式(把yes改为no)
4 88行: protected-mode no
5 3、重启服务
6 sudo /etc/init.d/redis-server restart
```

- 远程连接测试

Windows连接Ubuntu的Redis服务

```
1 # cmd命令行
2 1、e:
3 2、cd Redis3.0
4 3、redis-cli -h x.x.x.x -a 123456
5 4、x.x.x.x:6379>ping
```

数据类型

- 通用命令 适用于所有数据类型

```
1 # 切换库(number的值在0-15之间,db0 ~ db15)
2 select number
3 # 查看键
4 keys 表达式 # keys *
5 # 数据类型
6 TYPE key
7 # 键是否存在
8 exists key
9 # 删除键
10 del key
```

```
11 # 键重命名
12 rename key newkey
13 # 清除当前库中所有数据（慎用）
14 flushdb
15 # 清除所有库中所有数据（慎用）
16 flushall
```

字符串类型(string)

- 特点

```
1 1、字符串、数字，都会转为字符串来存储
2 2、以二进制的方式存储在内存中
```

字符串常用命令-必须掌握

```
1 # 1. 设置一个key-value
2 set key value
3 # 2. 获取key的值
4 get key
5 # 3. key不存在时再进行设置(nx)
6 set key value nx # not exists
7 # 4. 设置过期时间(ex)
8 set key value ex seconds
9
10 # 5. 同时设置多个key-value
11 mset key1 value1 key2 value2 key3 value3
12 # 6. 同时获取多个key-value
13 mget key1 key2 key3
```

字符串常用命令-作为了解

```
1 # 1. 获取长度
2 strlen key
3 # 2. 获取指定范围切片内容
4 getrange key start stop
5 # 3. 从索引值开始，value替换原内容
6 setrange key index value
7 # 4. 追加拼接value的值
8 append key value
```

数值操作-字符串类型数字(必须掌握)

```

1 # 整数操作
2 INCRBY key 步长
3 DECRBY key 步长
4 INCR key : +1操作
5 DECR key : -1操作
6 # 应用场景: 抖音上有人关注你了, 是不是可以用INCR呢, 如果取消关注了是不是可以用DECR
7 # 浮点数操作: 自动先转为数字类型, 然后再进行相加减, 不能使用append
8 incrbyfloat key step

```

键的命名规范

mset wang:email wangweichao@tedu.cn

```

1 127.0.0.1:6379> mset wang:email wangweichao@tedu.cn guo:email guods@tedu.cn
2 OK
3 127.0.0.1:6379> mget wang:email guo:email
4 1) "wangweichao@tedu.cn"
5 2) "guods@tedu.cn"
6 127.0.0.1:6379>

```

string命令汇总

```

1 # 字符串操作
2 1、 set key value
3 2、 set key value nx
4 3、 get key
5 3、 mset key1 value1 key2 value2
6 4、 mget key1 key2 key3
7 5、 set key value nx ex seconds
8 6、 strlen key
9 # 返回旧值并设置新值 (如果键不存在, 就创建并赋值)
10 7、 getset key value
11 # 数字操作
12 7、 incrby key 步长
13 8、 decrby key 步长
14 9、 incr key
15 10、 decr key
16 11、 incrbyfloat key number#(可为正数或负数)
17 # 设置过期时间的两种方式
18 # 方式一
19 1、 set key value ex 3
20 # 方式二
21 1、 set key value
22 2、 expire key 5 # 秒
23 3、 pexpire key 5 # 毫秒
24 # 查看存活时间
25 ttl key
26 # 删除过期
27 persist key

```

- string数据类型注意

```
1 # key取值原则
2 1、key值不宜过长，消耗内存，且在数据中查找这类键值的计算成本高
3 2、不宜过短，可读性较差
4 # 值
5 1、一个字符串类型的值最多能存储512M内容
```

练习

```
1 1、查看 db0 库中所有的键
2 # select 0
3 # keys *
4 2、设置键 trill:username 对应的值为 user001，并查看
5 # set trill:username user001
6 3、获取 trill:username 值的长度
7 # strlen trill:username
8 4、一次性设置 trill:password、trill:gender、trill:fansnumber 并查看（值自定义）
9 # mset trill:password 123 trill:gender M trill:fansnumber 500
10 5、查看键 trill:score 是否存在
11 # exists trill:score
12 6、增加10个粉丝
13 # incrby trill:fansnumber 10
14 7、增加2个粉丝（一个一个加）
15 # incr trill:fansnumber
16 # incr trill:fansnumber
17 8、有3个粉丝取消关注你了
18 # decrby trill:fansnumber 3
19 9、又有1个粉丝取消关注你了
20 # decr trill:fansnumber
21 10、思考、思考、思考...，清除当前库
22 # flushdb
23 11、一万个思考之后，清除所有库
24 # flushall
```

列表数据类型（List）

• 特点

```
1 1、元素是字符串类型
2 2、列表头尾增删快，中间增删慢，增删元素是常态
3 3、元素可重复
4 4、最多可包含 $2^{32}-1$ 个元素
5 5、索引同python列表
```

• 列表常用命令

```
1 # 增
2 1、从列表头部压入元素
3 LPUSH key value1 value2
4 2、从列表尾部压入元素
5 RPUSH key value1 value2
6 3、从列表src尾部弹出1个元素，压入到列表dst的头部
```

```

7   RPOPLPUSH src dst
8   4、在列表指定元素后/前插入元素
9   LINSERT key after|before value newvalue
10
11  # 查
12  5、查看列表中元素
13   LRANGE key start stop
14   # 查看列表中所有元素: LRANGE key 0 -1
15  6、获取列表长度
16   LLEN key
17
18  # 删
19  7、从列表头部弹出1个元素
20   LPOP key
21  8、从列表尾部弹出1个元素
22   RPOP key
23  9、列表头部,阻塞弹出,列表为空时阻塞
24   BLPOP key timeout
25  10、列表尾部,阻塞弹出,列表为空时阻塞
26   BRPOP key timeout
27   # 关于BLPOP 和 BRPOP
28   1、如果弹出的列表不存在或者为空,就会阻塞
29   2、超时时间设置为0,就是永久阻塞,直到有数据可以弹出
30   3、如果多个客户端阻塞再同一个列表上,使用First In First Service原则,先到先服务
31  11、删除指定元素
32   LREM key count value
33   count>0: 表示从头部开始向表尾搜索,移除与value相等的元素,数量为count
34   count<0: 表示从尾部开始向表头搜索,移除与value相等的元素,数量为count
35   count=0: 移除表中所有与value相等的值
36  12、保留指定范围内的元素
37   LTRIM key start stop
38   LRTIM mylist1 0 2 # 只保留前3条
39   # 应用场景: 保存微博评论最后500条
40   LTRIM weibo:comments 0 499
41
42  # 改
43  13、LSET key index newvalue

```

练习

```

1   1、查看所有的键
2   # keys *
3   2、向列表 spider:urls 中以RPUSH放入如下几个元素: 01_baidu.com、02_taobao.com、03_sina.com、
4   04_jd.com、05_xxx.com
5   # RPUSH spider:urls 01_xxx 02_xxx 03_xxx
6   3、查看列表中所有元素
7   # LRANGE spider:urls 0 -1
8   4、查看列表长度
9   # LLEN spider:urls
10  5、将列表中01_baidu.com 改为 01_tmall.com
11  # LSET spider:urls 0 01_tmall.com
12  6、在列表中04_jd.com之后再加1个元素 02_taobao.com
   # LINSERT spider:urls after 04_jd.com 02_taobao.com

```

```
13 7、弹出列表中的最后一个元素
14   # RPOP spider:urls
15 8、删除列表中所有的 02_taobao.com
16   # LREM spider:urls 0 02_taobao.com
17 9、剔除列表中的其他元素，只剩前3条
18   # LTRIM spider:urls 0 2
```

与python交互

- 模块(redis)

Ubuntu

```
1 | sudo pip3 install redis
```

Windows

```
1 # 方法1. python -m pip install redis
2 # 方法2. 以管理员身份打开cmd命令行
3     pip install redis
```

- 使用流程

```
1 import redis
2 # 创建数据库连接对象
3 r = redis.Redis(host='127.0.0.1',port=6379,db=0,password='123456')
```

- 通用命令代码示例

```
1 import redis
2
3 # 创建连接对象
4 r = redis.Redis(host='192.168.153.146',port=6379,db=0)
5
6 # r.keys('*') -> 列表
7 key_list = r.keys('*')
8 for key in key_list:
9     print(key.decode())
10
11 # b'list'
12 print(r.type('mylist'))
13 # 返回值: 0 或者 1
14 print(r.exists('spider:urls'))
15 # 删除key
16 r.delete('mylist2')
```

- python操作list

```
1 import redis
2
```



```

3 r = redis.Redis(host='192.168.153.146',port=6379,db=0)
4
5 # pylist: ['pythonweb','socket','pybase']
6 r.lpush('pylist','pybase','socket','pythonweb')
7 # pylist: ['spider','pythonweb','socket','pybase']
8 r.linsert('pylist','before','pythonweb','spider')
9 # 4
10 print(r.llen('pylist'))
11 # [b'spider', b'pythonweb', b'socket', b'pybase']
12 print(r.lrange('pylist',0,-1))
13 # b'pybase'
14 print(r.rpop('pylist'))
15 # [b'spider', b'pythonweb']
16 r.ltrim('pylist',0,1)
17
18 while True:
19     # 如果列表中为空时,则返回None
20     result = r.brpop('pylist',1)
21     if result:
22         print(result)
23     else:
24         break
25
26
27 r.delete('pylist')

```

list案例: 一个进程负责生产url, 一个进程负责消费url

进程1: 生产者

```

1 import redis
2 import time
3 import random
4
5 r = redis.Redis(host='192.168.153.146',port=6379,db=0)
6
7 # 生产者开始生产url地址
8 for page in range(0,67):
9     url = 'http://app.mi.com/category/2#page=%s' % str(page)
10    r.lpush('spider:urls',url)
11    time.sleep(random.randint(1,3))

```

进程2: 消费者

```

1 import redis
2
3 r = redis.Redis(host='192.168.153.146',port=6379,db=0)
4
5 while True:
6     # url: (b'spider:urls',b'http://xiaomixxx')
7     url = r.brpop('spider:urls',5)
8     if url:
9         print('正在抓取:',url[1].decode())
10    else:
11        print('抓取结束')
12        break

```

使用进程模块来实现试试?

```

1 import redis
2 import time
3 import random
4 from multiprocessing import Process
5
6 class Spider(object):
7     def __init__(self):
8         r = redis.Redis(host='192.168.153.146',port=6379,db=0)
9
10    def product(self):
11        # 生产者开始生产url地址
12        for page in range(0,67):
13            url = 'http://app.mi.com/category/2#page=%s' % str(page)
14            self.r.lpush('spider:urls',url)
15            time.sleep(random.randint(1,3))
16
17    def consumer(self):
18        while True:
19            # url: (b'spider:urls',b'http://xiaomixxx')
20            url = self.r.brpop('spider:urls', 5)
21            if url:
22                print('正在抓取:', url[1].decode())
23            else:
24                print('抓取结束')
25                break
26
27    def run(self):
28        p1 = Process(target=self.product)
29        p2 = Process(target=self.consumer)
30        p1.start()
31        p2.start()
32        p1.join()
33        p2.join()
34
35 if __name__ == '__main__':
36     spider = Spider()
37     spider.run()

```

- python操作Estring

```
1 import redis
2
3 r = redis.Redis(host='192.168.153.146',port=6379,db=0)
4
5 r.set('username','guods')
6 print(r.get('username'))
7 # mset参数为字典
8 r.mset({'username':'xiaoze','password':'123456'})
9 # 列表: [b'xiaoze', b'123456']
10 print(r.mget('username','password'))
11 # 6
12 print(r.strlen('username'))
13
14 # 数值操作
15 r.set('age','25')
16 r.incrby('age',10)
17 r.decrby('age',10)
18 r.incr('age')
19 r.decr('age')
20 r.incrbyfloat('age',3.3)
21 r.incrbyfloat('age',-3.3)
22 print(r.get('age'))
23
24 r.delete('username')
```