



链滴

测试人员都需要了解哪些 redis 知识

作者: [zyjImmortal](#)

原文链接: <https://ld246.com/article/1583338287797>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



在介绍redis之前，先来几个问题，

- 1、redis是什么，为什么使用redis会非常快
- 2、redis支持的数据类型都有那些
- 3、如何通过python与redis进行交互

redis为什么这么快

- 编写语言

redis是采用C语言编写的，好处就是底层代码执行效率高，依赖性低，没有太多运行时的依赖，而且系统的兼容性好，稳定性高

- 存储

redis是基于内存的数据库，可有避免磁盘IO，因此也被称作缓存工具

- 数据结构

redis采用key-value方式进行存储，也就是使用的hash结构进行操作，数据的操作时间复杂度是O(1)。

- 设计模型

redis采用的是单进程单线程的模型，可以避免上下文切换和线程之间引起的资源竞争。而且Redis还用了IO多路复用技术，这里的多路指的是多个socket网络连接，复用指的是一个线程中处理多个IO请求，这样可以减少网络IO的消耗，大幅度提升效率。

Redis数据类型

redis支持的value值的数据类型是比较多的，包括字符串、哈希、列表、集合、有序集合等。

- 字符串

字符串类型是redis提供的最基本的数据类型，对应的结构是key-value。比如说要设置某个键，使用方法是

set key value, 取值就是 get key

```
127.0.0.1:6379> set name zhangsan
OK
127.0.0.1:6379> get name
"zhangsan"
127.0.0.1:6379>
```

- hash

hash提供了字段和字段值的映射，对应的结构是key-field-value，设置某个键的hash值，可以使用

hset key field value

```
hset user username zhangfei
hset user age 45
```

如果同时想要将多个field-value设置给某个键key的时候，可以使用hmset key field value [field value ...]

```
hmset user username liubei age 99;
```

取值的时候使用hget key field，取键的多个field字段值的时候使用 hmget key field [field..]

```
127.0.0.1:6379> hmget user username age
1) "liubei"
2) "99;"
```

- 字符串列表 (list)

字符串列表这个数据类型底层是一个双向链表数据结构，可以向列表的两端添加数据，时间复杂度都O(1)的，也支持区间查询，想列表左侧添加元素使用：LPUSH key value [...]

```
127.0.0.1:6379> lpush cities beijing shanghai guangzhou
```

向列表右侧添加元素，RPUSH key value [...]

```
127.0.0.1:6379> rpush cities shenzhen tianjin
```

获取列表中某一段的内容，使用lrange key start stop

```
127.0.0.1:6379> lrange cities 0 4
1) "guangzhou"
2) "shanghai"
3) "beijing"
4) "shenzhen"
5) "tianjin"
```

- 字符串集合set

字符串集合是字符串类型的无序集合，与列表list的区别在于集合中的元素是无序的，同时元素是不重复的。

在集合中添加元素，可以使用sadd key member [...]

```
127.0.0.1:6379> sadd heros liubei guanyu zhangfei zhaoyun
(integer) 4
```

如果想要在集合中删除元素，使用SREM key member [...]

```
127.0.0.1:6379> srem heros liubie
(integer) 0
```

从集合中获取所有元素使用 smembers key

```
127.0.0.1:6379> smembers heros
1) "zhaoyun"
2) "zhangfei"
3) "guanyu"
4) "liubei"
```

判断元素是否存在集合中，SISMEMBER key member

```
127.0.0.1:6379> sismember heros zhaoyun
(integer) 1
```

- 有序字符串集合 (SortedSet, 简称 ZSET)

ZSET是set的升级版，在集合的基础上增加了分数属性，这个属性在添加修改元素的时候可以被指定，每次指定后，ZSET 都会按照分数来进行自动排序，也就是说我们在给集合 key 添加 member 的时候，可以指定 score。

列表 list 是通过双向链表实现的，在操作左右两侧的数据时会非常快，而对于中间的数据操作则相对慢。有序集合采用 hash 表的结构来实现，读取排序在中间部分的数据也会很快。同时有序集合可以通过 score 来完成元素位置的调整，但如果我们想要对列表进行元素位置的调整则会比较麻烦

在zset有序集合汇总中添加元素和分数，使用ZADD key score member [...]

```
127.0.0.1:6379> zadd userscore 3434 zhangfei 4455 liubei 6666 guanyu
(integer) 3
```

获取某个元素的分数，zscore key member

```
127.0.0.1:6379> zscore userscore zhangfei
"3434"
```

获取某个范围的元素列表,如果想要分数从小到大排序输出，使用zrange key start stop [WITHSCORES]，如果想要分数从大到小输出使用 ZREVRANGE key start stop [WITHSCORES]，WITHSCORES 个可选项，如果使用 WITHSCORES 会将分数一同显示出来。

```
127.0.0.1:6379> zrange userscore 0 3
1) "zhangfei"
2) "liubei"
```

```
3) "guanyu"
127.0.0.1:6379> zrange userscore 0 3 withscores
1) "zhangfei"
2) "3434"
3) "liubei"
4) "4455"
5) "guanyu"
6) "6666"
127.0.0.1:6379> zrevrange userscore 0 3 withscores
1) "guanyu"
2) "6666"
3) "liubei"
4) "4455"
5) "zhangfei"
6) "3434"
```

删除一个或多个元素，可以使用 ZREM key member [member ...]

```
127.0.0.1:6379> zrem userscore liubei
(integer) 1
127.0.0.1:6379> zrange userscore 0 3 withscores
1) "zhangfei"
2) "3434"
3) "guanyu"
4) "6666"
```

python 中使用redis

用python操作redis需要先安装redis的python库，pip install redis，python提供了两种连接redis方式，一种是直接连接，

```
r = redis.Redis(host='localhost', port= 6379)
```

另一种方式是使用连接池，

```
pool = redis.ConnectionPool(host='localhost', port=6379)
r = redis.Redis(connection_pool=pool)
```

连接池机制可以避免频繁创建和释放连接，提升整体的性能,这个机制可以让我们事先创建好多个连接将其放到连接池中，当我们需要进行 Redis 操作的时候就直接从连接池中获取，完成之后也不会直接放掉连接，而是将它返回到连接池中。

python操作redis和命令行操作差不多，这里就不多说了，大家可以自己尝试一下。