Redis

## 简介

Redis是开源内存data structure store，可以作为db\cache\message broker

redis客户端:StackExchange.Redis

客户端官网:https://github.com/StackExchange/StackExchange.Redis

redis官网:https://redis.io/

redis官方不支持windows(memcached也不支持,不建议使用memcached)

Microsoft open tech group支持win64版本地址:https://github.com/MicrosoftArchive/redis

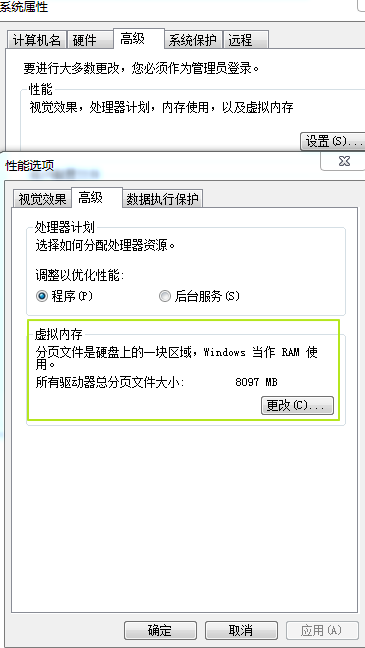
## 基础

## 部署

在Windows上部署Redis。

从Redis3.0开始，启动时需要最小量的heap，当需要时分配。Redis使用system page file来back redis heap。建议配置redis heap page file为两倍物理内存。

a safe minimum requirement for the system paging file is 2 times the size of physical memory. 默认Windows的配置满足需求，但可适当增加到3倍物理内存。默认的配置如下：



注：其他程序也使用page file。

### 配置

Redis可以安装为service或者独立运行。提供两种安装包



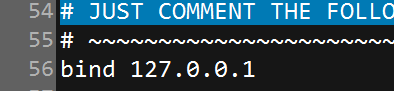
#### 独立运行

使用版本Redis-x64-3.2.100

使用redis.windows.conf配置文件，使用如下命令运行

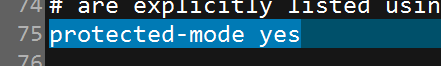
redis-server.exe redis.windows.conf

1、注释掉bind，这样将绑定所有IP



2、禁用保护模式，保护模式下，只能本机连接或者必须设置密码

protected-mode no#禁用保护模式



3、修改端口，默认使用端口6379



4、tcp-backlog默认可不修改



5、配置timeout，close the connection after a client is idle for N seconds

建议保留0



6、配置tcp keepalive，多长时间发送ack，建议60



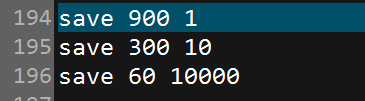
7、日志级别，配置成notice或者warn



8、配置number of databases



9、save <seconds> <changes> Will save the DB if both the given number of seconds and the given number of write operations against the DB occurred. 默认可不修改



10、禁用stop-writes-on-bgsave-error 修改为no



##### 复制和主从模式

主从模式的好处是从是可以读的，但是从一般是不可以写的。

1）redis replication是异步的。

2）replication是原子的和自动的。可以自动重连和重新同步主服务器

# slaveof <masterip> <masterport>

1、slave在和master丢失连接时，仍然响应客户端请求



2、配置slave只读，默认是只读的

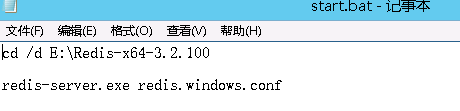


3、

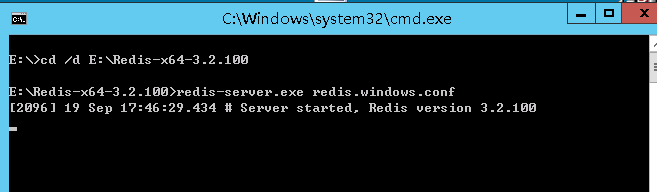
下面是一个配置好的，utf8无bom



创建bat文件



启动后



#### 安装为服务

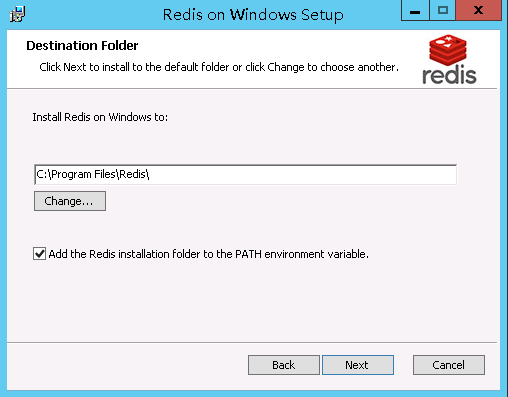
使用MSI安装包安装时，Redis将会被安装成服务。如果需要改变配置，修改redis.windows-service.conf文件，重启服务后配置生效。

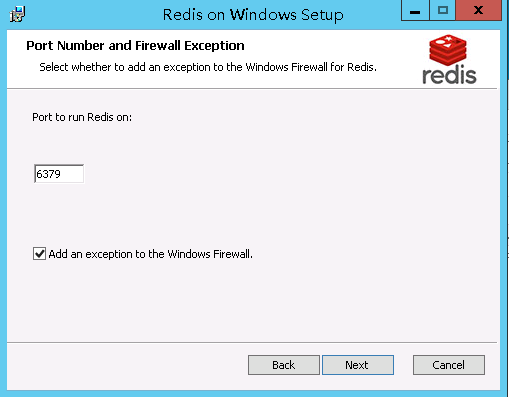
Run -\> services.msc -\> Redis -\> Restart



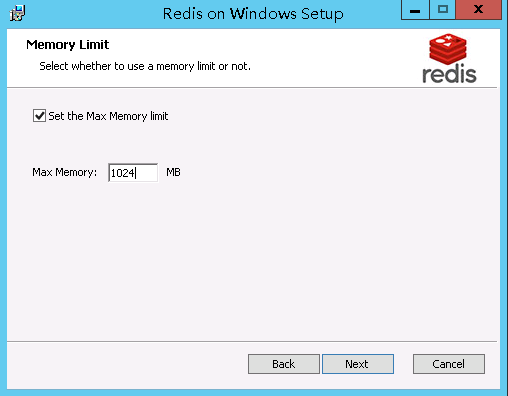


目录最好选择D盘

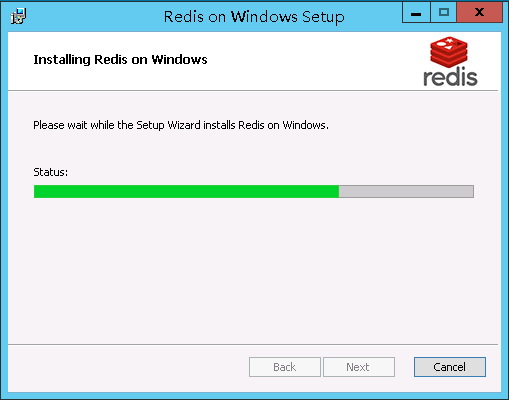




redis的maxmemory参数用于控制redis可使用的最大内存容量。如果超过maxmemory的值，就会动用淘汰策略来处理expaire字典中的键。









##### 修改redis.windows-service.conf

1. 修改bind，监听所有的IP，注释掉bind



1. 禁用protected mode，否则只能本地连接



1. 修改监听端口 默认是6379



1. 修改tcp-backlog，默认511，可不修改，表示客户端连接排队的队列长度



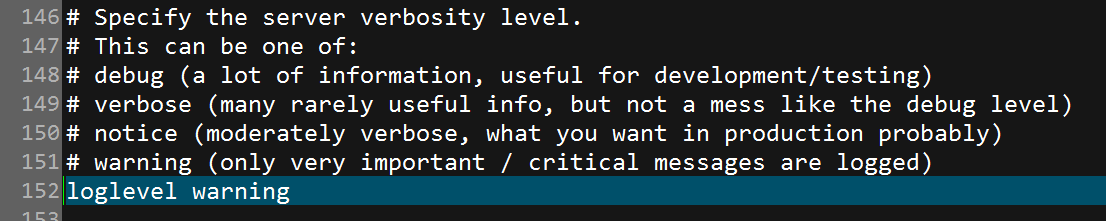
1. 配置timeout 如果客户端闲置N秒后关闭，修改为127



1. 配置tcp-keepalive为60



1. 修改日志级别



1. 指定日志文件名



1. 禁用syslog

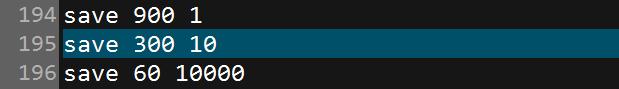


1. 指定the number of databases



1. 保存db到disk save <seconds> <changes>

指定多长时间内有多少次修改，保存到数据库



1. 禁用stop-writes-on-bgsave-error



1. 查看工作目录



###### 主从复制

1. 主从复制 使用slaveof使本实例成为masterip的slave



注：Redis replication是异步的。复制在断线后可以自动重试连接和重新同步

1. 配置Slave在与master断开连接好，仍然响应客户端请求



1. 配置slave只读，不接受写



1. 配置复制的策略 disk or socket

Slave在断线后需要full sync（全表同步），从master上传递一个rdbfile到slave,有两种方式：1)disk-backed:master复制rdb文件，然后传输，2)diskless:master直接传递rdb文件。

建议使用disk-backed。



1. 配置slave发送PINGS到server的时间间隔。 单位是秒



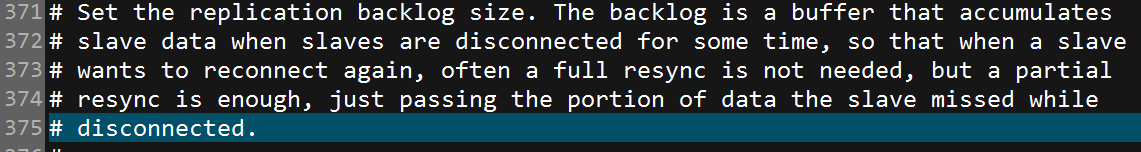
1. 配置repl 超时时间 timeout



1. Disable tcp nodelay 配置成yes



1. 配置 replication backlog size。





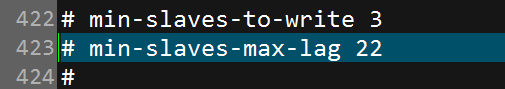
1. 配置slave断线多长时间释放backlog,单位秒



1. Slave优先级，低的数值认为优先级好，当master挂掉后，优先选择为新的master

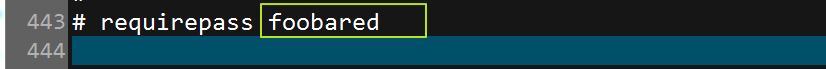


1. 配置master停止接收写操作，如果少于N个slave和lag小于等于M秒。Lag从slave中最近一次平计算的时间

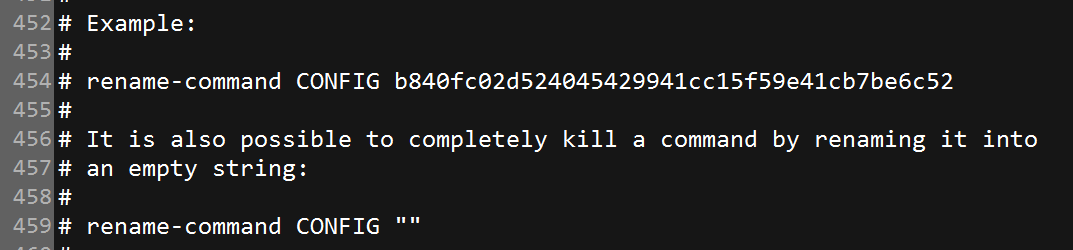


###### 安全性

1. 配置redis的连接密码



1. 命令重命名



1. 限制同时连接的最大客户端数



1. 配置redis是否使用持久还是只使用内存，在windows上关闭持久，性能会有较大提升，不会导致fork调用。



1. 设置redis占用的最大内存



1. 设置当maxmemory到达时，redis如何清理内存



1. 配置清理检测样本



###### Redis 集群

1. 普通的redis节点不能成为redis cluster集群节点，需要配置如下选项



1. 每个集群节点有一个集群配置文件，该文件不能手动编辑，由redis集群节点创建和更新



1. 集群节点超时多少毫秒认为是断线，单位毫秒



1. cluster-slave-validity-factor



1. Slow log单位微秒，记录执行命令超过指定时间，记录下来



###### 事件通知

Redis can notify pub/sub clients about events happening in the key space.

默认是全部disabled的。

下面是一个配置好的单机文件(Redis-x64-3.2.100), 编码ANSI

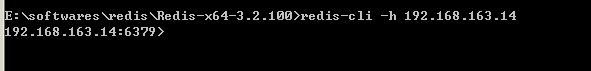


## Redis Command

命令行客户端

Redis-cli --help

Redis-cli –h hostname –p port #-h指定服务器 –p指定端口



依次是命令列举常用的，不全。

### APPEND

Append key value

Time O(1)

如果键存在，并且值是字符串，将value添加到字符串结尾。如果键不存在，创建空字符串然后添加到字符串结尾。返回字符串的长度

### AUTH

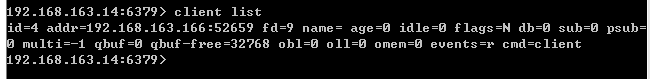
AUTH password

验证redis server

### CLIENT LIST

TIME O(n) n是客户端连接数

返回当前所有的客户端连接信息



### CLIENT GETNAME

返回当前连接的名字或者null（如果没有设置） ，该名字可以通过CLIENT SETNANE设置

TIME O(1)

### CLIENT SETNAME

TIME O(1)

设置当前连接的name

### DEL key [key …]

Time O(n) n是del后key的个数

移除指定的keys，如果不存在则忽略。

返回溢出key的数量

### EXISTS key [key …]

TIME O(1)

判断key是否存在，返回存在key的数量

### EXPIRE key seconds

TIME O(1)

设置key的超时时间，超时时间过后，key自动删除。如果超时时间已经存在，更新为新的超时时间。

所有的变更操作不会更改超时时间，替换操作有可能更改超时时间

返回1 ，设置成功， 0如果键不存在

### EXPIREAT key timestamp

TIME O(1)

与expire相同，除了指定的是 unix timestamp （从1970.1.1经过的秒数）。

### FLUSHALL [ASYNC]

TIME O(N) n是数据库键的数量

删除所有数据库的所有键

ASYNC指定异步方式

RETURN VALUE simple string reply

### FLUSHDB [ASYNC]

删除当前选中DB的所有的键。

TIME O(N) n是db中所有的键的个数

RETURN VALUE simple string reply

### GET key

TIME O(1)

获取key关联的值，如果不存在返回nil.如果存储的值不是string类型，返回error

get只能用来获取字符串值

### GETRANGE key start end

Time O(n) n是返回字符串的长度

返回key关联的值的start到end部分，从0开始计数，同时包含start和end。

### GETSET key value

Time时间复杂度 O(1)

设置key到value并且返回存在key里的旧值。返回error如果key exists but not hold a string value.

### HDEL key field [field …]

TIME O(n) n指要移除的field的个数

根据key移除指定的field from hash。如果key不存在，什么也不做返回0

如果field不存在，忽略该field.

返回移除field的个数。

### HEXISTS key field

TIME O（1）

检测key关联的hash是否有指定的field

返回1如果存在，0如果不存在

### HGET key field

TIME O(1)

返回field关联的值 在key关联的hash中

如果不存在，返回nil

HGETALL key

TIME O(N) n是hash的size

返回key关联的hash的所有fields和values

返回值:  list of fields and their values stored in the hash, or an empty list when key does not exist.

### HEKYS key

TIME O(N) n是hash的size

返回key关联的hash的所有filed，or empty list if key not exist

### HLEN key

TIME O(1)

返回number of fields in the hash, or 0 when key does not exist.

### HMGET key field [field …]

TIME O（n） n指请求的field的数量

返回fields关联的value存储在指定的key

对于field不存在的，返回nil，不存在的key，认为是空hash，返回一个list of nil.

### HMSET key field value [field value …]

TIME O（n） n是需要设置的field的个数

设置key关联的hash中设置指定的field value对。

如果key不存在，创建hash。如果field存在，则覆盖field。

### HSET key field value

TIME O（1）

设置field in hash at key to value. 如果key不存在，创建key关联的hash。如果field存在，重写。

返回1，如果field是新增 返回0如果field是更新。

### HSETNX key field value

TIME : O（1）

如果field不存在，则创建并设置value。如果key不存在，创建。如果存在field，什么也不做。

返回：1 如果设置成功 0如果field已经存在

### HSTRLEN key field

TIME : O（1）

返回关联的字符串的长度，如果没有返回0

### HVALS key

TIME : O（n） n is the size of hash

返回key关联的hash的所有的值，如果不存在key返回empty list

### INCR key

TIME : O（1）

操作限于64位有符号数。key关联的数字加1，如果不存在，则设置为0后加1.如果类型不对或者字符串不能转换为整数，返回错误。

redis没有数字类型。这是一个字符串操作。字符串转换为数字是base10的64为有符号证书。

### INCRBY key increment

Time: O（1）

增加key关联的value加increment。如果key不存在，设置为0后增加，如果类型不对或者字符串不能转换为整数，返回错误。

redis没有数字类型。这是一个字符串操作。字符串转换为数字是base10的64为有符号证书。

### KEYS pattern

TIME : O(n) n是db中的key的数量

返回满足pattern的所有keys

在产品中谨慎使用该命令。因为是O(N)的。

?配置一个字符

\*配置0个或者多个

### LINDEX key index

TIME : O（n） n is the number of elements to traverse to get to the element at index. Get first or last element of the list is O(1)

返回element存储在key的list中第index索引的项。是从0开始索引，0是指第一个元素。-1是指最后一个元素。如果key不是list，返回error。

### LINSERT key BEFORE|AFTER pivot value

TIME: O（n） n是在遍历到pivot之前遍历的元素的个数。从左边开始遍历，因此右边结尾是O(n)的。

在key索引的list中在pivot之前或者之后插入值value。如果key不存在，什么也不做。如果key索引的不是list，返回一个error。如果pivot找不到，什么也不做。

返回list的长度如果插入成功，-1如果pivot没有找到。

### LLEN key

TIME : O（1）

返回存储在key的list的长度，如果key不存在，返回0.如果key关联的类型不是list，返回error。

### LPOP key

TIME : O（1）

移除并返回key索引的list的第一个元素。Nil如果key不存在

### LPUSH key value [value …]

Time : O（1）

插入所有的value到key索引的list的头部。如果key不存在，创建新的empty list之后插入。如果key索引的不是list，返回error

返回插入操作之后的list的长度

### LPUSH key value

TIME： O(1)

在key索引的list的头部添加value。仅当key存在并且是list时，操作，否则什么也不做。

返回操作之后list的长度。

### LRANGE key start stop

TIME : O（s+n） s指的是起始位置，n指的是range的长度。

返回key存储的list中指定范围的元素。Start和stop是0索引的，0是第一个元素

包含开头和结尾。

### LREM key count value

O（n）n是list的长度

删除等于value的前count个元素。

Count>0从头到尾搜索 count<0从尾到头搜索 count=0删除所有等于value的

返回移除元素的个数

### LSET key index value

O(N) where N is the length of the list. Setting either the first or the last element of the list is O(1).

设置list element at index to value。

### LTRIM key start stop

Time:O（n） n是指要移除元素的个数

收缩list使其只有指定范围的的元素。Start和end是0 based。

### MGET key [key …]

O（n） n是要获取key的个数

返回元获取的key的所有的value。如果key不存在或者not hold string，返回nil。

### MSET key value [key value …]

O（n） n是指要设置元素的个数

设置多个key value，如果key存在，则进行替换。

### MSETNX key value [key value…]

O（n） n是要设置key的个数

如果有一个key存在，则所有的key都不设置并返回0，否则进行设置并返回1

### PERSIST key

TIME : O（1）

移除key上的timeout，使key不会失效

返回1，如果timeout移除，返回0如果key不存在或者没有timeout

### PEXPIRE key milliseconds

TIME: O（1）

设置超时时间，单位毫秒。1如果timeout设置成功，0如果key不存在。

### PEXPIREAT key milliseconds-timestamp

TIME： O(1)

类似于PEXPIRE，但是使用的自1970年1月1日的毫秒数。

1如果timeout设置成功，0如果key不存在。

### PING [message]

如果没有参数，返回pong，否则返回message。用来测试连接

### PSETEX key milliseconds value

Time : O（1）

设置key为指定的value，并且带有过期时间 毫秒为单位。

### PTTL key

TIME ： O(1)

返回key剩余的生存毫秒数。返回-1如果key不存在或者没有过期时间。

### QUIT

关闭连接

### RENAME key newkey

将key重名为newkey。如果key不存在，返回error。

如果newkey存在，value会被覆盖。

### RENAMENX key newkey

time： O(1)

如果newkey不存在，将key重新命名为newkey。如果key不存在或者newkey存在返回错误

### RPOP key

TIME : O（1）

移除并返回key索引的list的最后一个元素。 如果不存在返回nil

### RPUSH key value [value …]

time： O(1)

插入所有指定的值到key索引的list的结尾。

如果key索引的list不存在，创建一个空的list后操作。如果不是list，返回错误。

### RPUSHX key value

TIME ： O(1)

将value放到key索引的list的结尾。只有在key已经存在并且hold a list下执行，否则什么也不做。

返回操作后，list的长度。

### SADD key member [member …]

TIME : O（1）对于每个要添加的元素

将members添加到key索引的set中。如果key不存在，创建新的set。如果key存在，并且不是set返回error

set是无序集合，因此重复的member将会被忽略。

### SCARD key

返回集合中元素数。如果key不存在返回0

### SDIFF key [key …]

O（n） n是所有集合的所有的元素数

返回集合间的差，如果不存在某个key，则认为是empty set

### SDIFFSTORE destination key [key …]

计算集合的差，并将其保存到destination中，如果destination已经存在则覆盖。

### SET key value [EX seconds] [PX milliseconds] [NX|XX]

TIME : O(1)

将key设置为string value。如果key存在，则覆盖。 先前设置的超时被忽略。

EX 设置指定的超时时间单位秒

PX 设置指定的超时时间单位毫秒

NX 设置key仅当key不存在 XX 设置key仅当其已经存在。

### SETEX key seconds value

TIME： O(1)

设置key到string value，并带有timeout。

### SETNX key value

TIME ： O(1)

设置key到string value,仅当key不存在时。如果已经存在，什么也不做。

1如果key设置，0如果key不存在

### SINTER key [key …]

TIME： O(n\*m) n是最小的集合的元素数M是集合的个数

返回集合的交。

### SINTERSTORE destination key [key …]

TIME： O(n\*m) n是最小的集合的元素数M是集合的个数

返回集合的交，将其设置到destionation，如果destination存在，则overwritten。

### SISMEMBER key member

TIME：O(1)

判断member是否在key索引的set中

1如果在集合中， 0如果不存在。

### SMEMBERS key

TIME ： O(n) n是集合元素的个数

Return all elements of the set

### SMOVE source destination member

TIME： O(1)

将member从source集合移动到destination集合。如果source不存在或者不包含指定元素，什么也不做，返回0。返回error，如果source或者destination不是set。

### SORT key [BY pattern] [LIMIT offset count] [GET pattern [GET pattern …]] [ASC|DESC] [ALPHA] [STORE destination]

TIME: O（n+m\*logm） n是list或者set中元素的个数。M是返回元素的个数。

排序key关联的list、set和sorted set。默认是将string转换为numeric的double来计算。

如果要按照字母序来排序使用ALPHA。 ASC|DESC是升序还是降序。Redis是UTF8 ware的。

LIMIT 限制返回的数量，offset是从0开始计数的。Count指定返回的数量。

SORT mylist LIMIT 0 5 ALPHA DESC

### SPOP key [count]

TIME：O(1)

删除并返回一个或者多个随机元素存储在key的set。

### SRANDMEMBER key [count]

O（n） n是count

随机返回count个key关联的set的元素。

### SREM key member [member …]

TIME： O(n) n是要移除的members

移除key中set的元素。如果key不存在，返回0.如果key不是set，返回error。否则返回移除元素的个数。

### STRLEN key

TIME：O(1)

返回key关联的字符串的长度。返回error，如果key关联的不是字符串。如果key不存在，返回0.

### SUNION key [key …]

Time：O(n) n是keys关联的集合的所有元素。

返回所有key集合的并

### SUNIONSTORE destination key [key …]

Time：O(n) n是keys关联的集合的所有元素。

返回所有key集合的并的元素数，并且将集合的并保存到destination

### TIME

O（1）

返回当前服务器时间:two item list : a Unix timestamp and the amount of microseconds already elapsed in the current second.

### TTL key

time： O(1)

返回key剩余的live time。单位秒。 返回-1，如果没有超时，返回-2，如果key不存在

### TYPE key

time： O(1)

返回key关联的类型，如果没有key，返回none。

如：string, list, set, zset and hash.

### ZADD key [NX|XX] [CH] [INCR] score member [score member …]

每一个添加的元素是O(lgn)的

在key关联的zset添加关联score的member。如果member已经存在，socre更新并且位置也更新。如果key不存在，新建sorted set。如果key关联的不是zset，返回error。返回新增元素。score是字符串表示的double number

XX:仅仅更新member，不过不存在，也不添加

NX:不更新，仅仅添加

CH:返回更新元素而不是新增元素。

### ZCARD key

Time : O（1）

返回key中元素的个数。

### ZCOUNT key min max

O（n）n是zset中元素的个数。

返回key关联的zset中score在[min,max]中元素的数目

### ZINCRBY key increment member

Time: O（lgn）

Increments the score of member in the sorted set stored at key by increment. 如果member不存在，相当于新增。如果key不存在，新建zset。如果key关联的不是zset，返回error。score是字符串表示的double number

### ZINTERSTORE destination numkeys key [key …] [WEIGHTS weight [weight …]] [AGGREGATE SUM|MIN|MAX]

TIME： O(N\*K)+O(M\*log(M)) worst case with N being the smallest input sorted set, K being the number of input sorted sets and M being the number of elements in the resulting sorted set.

Computes the intersection of numkeys sorted sets given by the specified keys, and stores the result in destination. It is mandatory to provide the number of input keys (numkeys) before passing the input keys and the other (optional) arguments.

By default, the resulting score of an element is the sum of its scores in the sorted sets where it exists.

### ZLEXCOUNT key min max

TIME：O(logn) n是set中元素的数目。

返回the number of elements in the specified score range.

### ZRANGE key start stop [WITHSCORES]

**Time complexity:** O(log(N)+M) with N being the number of elements in the sorted set and M the number of elements returned.

Returns the specified range of elements in the sorted set stored at key.

### ZRANGEBYLEX key min max [LIMIT offset count]

TIME：O(log(N)+M) with N being the number of elements in the sorted set and M the number of elements being returned.

When all the elements in a sorted set are inserted with the same score, in order to force lexicographical ordering, this command returns all the elements in the sorted set at key with a value between min and max.

## Redis数据类型

Redis不限于string key和string value. Redis中 value不限于string。可以支持如下数据结构：

* Binary-safe strings
* Lists:collections of string sorted by the order by insertion。 Redis的list是double linked list.
* Sets:unique,unsorted string elements
* Sorted sets: similar to sets by every string element is associated to a floating number.并且按float number排序
* Hashes: fields 关联着 values。field和value都是string类型的。
* Bit arrays: handle String values like an array of bits
* HyperLogLogs:概率数据结构 used to estimate the cardinality of a set.

memcached中只有string类型

Redis key是binary safe的，可以是任意binary序列。 最长可以512M.建议只使用string作为key。 推荐使用格式：u:1000:pwd，key中不能有空格换行等字符。

Redis list是Linked list.