命令使用:

redis-cli [OPTIONS] [cmd [arg [arg ...]]]

选项说明:

-h <hostname> Server hostname (default: 127.0.0.1). ip地址

-p <port> Server port (default: 6379).　服务器端口号

-s <socket> Server socket (overrides hostname and port).

-a <password> Password to use when connecting to the server.　密码

-u <uri> Server URI.　url格式的地址

-r <repeat> Execute specified command N times.

-i <interval> When -r is used, waits <interval> seconds per command.

It is possible to specify sub-second times like -i 0.1.

-n <db> Database number.　指定数据库

-x Read last argument from STDIN.

-d <delimiter> Multi-bulk delimiter in for raw formatting (default: \n).

-c Enable cluster mode (follow -ASK and -MOVED redirections).

--raw Use raw formatting for replies (default when STDOUT is

not a tty).

--no-raw Force formatted output even when STDOUT is not a tty.

--csv Output in CSV format.

--stat Print rolling stats about server: mem, clients, ...　统计数据　连续输出

--latency Enter a special mode continuously sampling latency.

If you use this mode in an interactive session it runs

forever displaying real-time stats. Otherwise if --raw or

--csv is specified, or if you redirect the output to a non

TTY, it samples the latency for 1 second (you can use

-i to change the interval), then produces a single output

and exits.　延时统计

--latency-history Like --latency but tracking latency changes over time.

Default time interval is 15 sec. Change it using -i.

--latency-dist Shows latency as a spectrum, requires xterm 256 colors.

Default time interval is 1 sec. Change it using -i.

--lru-test <keys> Simulate a cache workload with an 80-20 distribution.

--replica Simulate a replica showing commands received from the master.

--rdb <filename> Transfer an RDB dump from remote server to local file. 导出rdb文件

--pipe Transfer raw Redis protocol from stdin to server.

管道模式

--pipe-timeout <n> In --pipe mode, abort with error if after sending all data.

no reply is received within <n> seconds.

Default timeout: 30. Use 0 to wait forever.

管道超时

--bigkeys Sample Redis keys looking for big keys.

--hotkeys Sample Redis keys looking for hot keys.

only works when maxmemory-policy is \*lfu.

--scan List all keys using the SCAN command.获取服务器所有的键

--pattern <pat> Useful with --scan to specify a SCAN pattern.

正则表达式　用于scan命令中

--intrinsic-latency <sec> Run a test to measure intrinsic system latency.

The test will run for the specified amount of seconds.

--eval <file> Send an EVAL command using the Lua script at <file>.

--ldb Used with --eval enable the Redis Lua debugger.

--ldb-sync-mode Like --ldb but uses the synchronous Lua debugger, in

this mode the server is blocked and script changes are

not rolled back from the server memory.

--cluster <command> [args...] [opts...]

Cluster Manager command and arguments (see below).

--verbose Verbose mode.

--no-auth-warning Don't show warning message when using password on command

line interface.

注意：

-u 选项中url格式参考文档https://www.iana.org/assignments/uri-schemes/prov/redis

格式为：redis://user:secret@localhost:6379/0?foo=bar&qux=baz

举例：

root@hylaz:~# redis-cli

127.0.0.1:6379> set name hylaz

OK

127.0.0.1:6379> quit

root@hylaz:~# redis-cli -h 127.0.0.1

127.0.0.1:6379> get name

"hylaz"

127.0.0.1:6379> select 6

127.0.0.1:6379[6]>

root@hylaz:~# redis-cli -h 127.0.0.1 -p 6379 -n 2

127.0.0.1:6379[2]> get age

server中统计选项

root@hylaz:~# redis-cli --stat

------- data ------ --------------------- load -------------------- - child -

keys mem clients blocked requests connections

11 835.52K 1 0 12 (+0) 5

11 835.52K 1 0 13 (+1) 5

11 835.52K 1 0 14 (+1) 5

11 835.52K 1 0 15 (+1) 5

列表中选项说明：

选项 含义

keys server中key的数量

mem 键值对的总内存量

clients 当前连接的总clients数量

blocked 当前阻塞的客户端数量

requests 服务器请求总次数 (+1) 截止上次请求增加次数

connections 服务器连接次数

使用info命令获取服务器的信息

导入rdb文件　命令：redis-cli --rdb rdb.log

root@hylaz:~# redis-cli --rdb rdb.log

SYNC sent to master, writing 344 bytes to 'rdb.log'

Transfer finished with success.

该命令选项实现：

向server发送SYNC命令，返回需要写的总字节数

从server读取总字节数据写到指定文件中

找出各种数据类型的最大键值对　命令：redis-cli --big-keys

root@hylaz:~# redis-cli --bigkeys

# Scanning the entire keyspace to find biggest keys as well as

# average sizes per key type. You can use -i 0.1 to sleep 0.1 sec

# per 100 SCAN commands (not usually needed).

[00.00%] Biggest string found so far 'name1' with 5 bytes

[00.00%] Biggest set found so far 'myset' with 1 members

[00.00%] Biggest string found so far 'key' with 6 bytes

-------- summary -------

Sampled 13 keys in the keyspace!

Total key length in bytes is 52 (avg len 4.00)

Biggest string found 'key' has 6 bytes

Biggest set found 'myset' has 1 members

12 strings with 33 bytes (92.31% of keys, avg size 2.75)

0 lists with 0 items (00.00% of keys, avg size 0.00)

1 sets with 1 members (07.69% of keys, avg size 1.00)

0 hashs with 0 fields (00.00% of keys, avg size 0.00)

0 zsets with 0 members (00.00% of keys, avg size 0.00)

0 streams with 0 entries (00.00% of keys, avg size 0.00)

该选项实现：通过使用scan命令遍历server中的键值对，针对不同数据类型进行统计，

找出server中热点key 命令：redis-cli --hotkeys

# Scanning the entire keyspace to find hot keys as well as

# average sizes per key type. You can use -i 0.1 to sleep 0.1 sec

# per 100 SCAN commands (not usually needed).

[00.00%] Hot key 'dd' found so far with counter 4

[00.00%] Hot key 'myset' found so far with counter 5

[00.00%] Hot key 'a' found so far with counter 5

[00.00%] Hot key 'dds' found so far with counter 4

[71.43%] Hot key 'aa' found so far with counter 4

[71.43%] Hot key 'key' found so far with counter 4

-------- summary -------

Sampled 14 keys in the keyspace!

hot key found with counter: 5 keyname: myset

hot key found with counter: 5 keyname: a

hot key found with counter: 4 keyname: dd

hot key found with counter: 4 keyname: dds

hot key found with counter: 4 keyname: aa

hot key found with counter: 4 keyname: key

选项实现：

1. redis实现8种缓存淘汰策略：

voltile-lru：从已设置过期时间的数据集（server.db[i].expires）中挑选最近最少使用的数据淘汰

volatile-ttl：从已设置过期时间的数据集（server.db[i].expires）中挑选将要过期的数据淘汰

volatile-random：从已设置过期时间的数据集（server.db[i].expires）中任意选择数据淘汰

volatile-lfu: 从已设置过期时间的数据集驱逐使用频率最少的键

allkeys-lru:从数据集（server.db[i].dict）中挑选最近最少使用的数据淘汰

allkeys-lfu: 从所有键中驱逐使用频率最少的键

allkeys-random：从数据集（server.db[i].dict）中任意选择数据淘汰

no-enviction（驱逐）：禁止驱逐数据 当内存不足以容纳新写入数据时，新写入操作会报错

需要设置淘汰策略为lru或者lfu

2. 命令实现使用scan命令遍历所有的键值对，针对每个键值对使用OBJECT freq 获取该键值对的信息

————————————————

版权声明：本文为CSDN博主「戴国进」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/JineD/article/details/124037249