

# Twemproxy 新增功能说明书

## 新增命令：replace\_server

**作用：**替换一个 server pool 中的 server，并更新配置文件。

**使用方法：**replace\_server arg1 arg2

**参数说明：**arg1 是被替换 server 的 ip:port；arg2 是新 server 的 ip:port

**示例：**replace\_server 10.101.1.10:22122 10.101.1.11:22122

**局限：**目前仅支持 redis:true。

## 日志切割功能

**作用：**把 twemproxy 日志切割成指定大小，并保留一定数量的日志文件。

**使用方法：**在 nutcracker 的启动项中增加了三个选项：-R(--log-rotate) -M(--log-file-max-size) -C(--log-file-count)。

**参数说明：**

-R 选项用于启用日志切割功能。默认不开启。

-M 选项可以指定一个数字来设置每个 log 文件的大小，默认单位是 byte，这个数字后面可以额外添加单位（目前仅支持 B，M，G，MB，GB，不区分大小写）。默认值为 1GB。

-C 可以指定一个数字 n 来设置最多可以保留 n 个 log 文件，n 可以为 -1, 0 和正整数。如果 n=-1，则保留所有 log 文件；如果 n=0，只保留当前 log 文件；如果 n>0，除当前 log 文件外，额外保留 n 个 log 文件。额外的 log 文件命名规则：当前 log 文件名\_切割时间。默认值为 10。

-M 和 -C 选项都是在 -R 选项启用的时候才有效。

**示例：**nutcracker -d -o nutcracker.log -R -M 20MB -C 2

```
-rw-r--r-- 1 root root 9.2M Jan 28 11:18 nutcracker.log
-rw-r--r-- 1 root root 21M Jan 28 11:18 nutcracker.log_2015-01-28_11.18.55.151
-rw-r--r-- 1 root root 21M Jan 28 11:18 nutcracker.log_2015-01-28_11.18.57.176
```

**局限：**选项 -C 目前最大值可以指定的值为 200。

## tcpkeepalive

**作用：**增加 Keep-alive 的机制可以检测并删除 tcp 死连接。

**使用方法：**yml 配置文件中增加了四个参数：tcpkeepalive，tcpkeepidle，tcpkeepcnt 和 tcpkeepintvl。

### 参数说明：

tcpkeepalive 参数决定是否开启 keepalive 机制。两个可选值：true|false。

tcpkeepidle 参数指定 tcp 连接多长时间空闲后检查时候该连接是否已死。单位为秒。

tcpkeepcnt 参数指定如果客户端一直没有回应，twemproxy 要持续几次探测行为。

tcpkeepintvl 参数指定 twemproxy 开始检查连接后，隔多久进行一次探测。单位为秒。

tcpkeepalive 为 true 时，其他三个参数才有效。

### 示例：

twem1:

```
listen: 127.0.0.1:22122
hash: fnv1a_64
hash_tag: "{}"
distribution: ketama
auto_eject_hosts: true
timeout: 400
redis: false
tcpkeepalive: true
tcpkeepidle: 800
tcpkeepcnt: 3
tcpkeepintvl: 60
server_connections: 5
servers:
- 127.0.0.1:23401:1 server1
- 127.0.0.1:23401:1 server2
- 127.0.0.1:23401:1 server3
```

局限：暂无。

## 管理功能

作用：增加 twemproxy 的管理接口。

使用方法： 开启一个端口，专门用于用户登陆管理 twemproxy。在 nutcracker 的启动项中增加了两个选项：-A (--proxy-adm-addr) -P (--proxy-adm-port)。

用户使用 telnet 登陆，即可以执行相应的新增命令（可通过 help 命令查看）：show\_conf，show\_oconf，show\_pools，show\_pool，show\_servers，find\_key，find\_keys，reload\_conf，set\_watch，del\_watch，reset\_watch，show\_watch。

### 参数说明：

-A 选项用于指定管理接口监听的 ip 地址，默认为 0.0.0.0。

-P 选项用于指定管理接口监听的 port 端口号，默认为 0，不开启。

### 命令说明:

show\_conf: 查看当前使用的配置项。不带参数。

show\_oconf: 查看上一次使用的配置项。不带参数。

show\_pools: 查看当前所有 pool 的名字。不带参数。

show\_pool: 查看某个 pool 的详细信息。带一个参数 (pool 的名字)。

show\_servers: 查看某个 pool 中包含的所有 server 信息。带一个参数 (pool 的名字)。

find\_key: 获取某个 key 所在的 server 信息。带两个参数 (pool 的名字和要查找的 key)。

find\_keys: 获取某些 key 所在的 server 信息。带一些参数 (pool 的名字和所有要查找的 key)

reload\_conf: 重新加载配置文件。

set\_watch: 在 zookeeper 设置一个 watch。目前仅支持 conf watch。

del\_watch: 在 zookeeper 删除一个 watch。目前仅支持 conf watch。

reset\_watch: 在 zookeeper 重新设置一个 watch。目前仅支持 conf watch。

show\_watch: 在 zookeeper 查看一个 watch。目前仅支持 conf watch。

示例: nutcracker -d -o nutcracker.log -A 127.0.0.1 -P 32045

```
help
*****
COMMAND : show_conf
DESCRIBE : display the conf file
USAGE    : no args
*****
COMMAND : show_oconf
DESCRIBE : display the old conf file if conf reloaded
USAGE    : no args
*****
COMMAND : show_pools
DESCRIBE : display all pools name
USAGE    : no args
*****
COMMAND : show_pool
DESCRIBE : display one pool's info
USAGE    : show_pool poolname
*****
COMMAND : show_servers
DESCRIBE : display one pool's servers info
USAGE    : show_servers poolname
*****
COMMAND : find_key
DESCRIBE : display a server which the key is on
USAGE    : find_key poolname key
*****
COMMAND : find_keys
DESCRIBE : display servers which the keys are on
USAGE    : find key poolname key1 key2 ...
*****
COMMAND : reload_conf
DESCRIBE : reload the nutcracker conf file
USAGE    : no args
*****
COMMAND : set_watch
DESCRIBE : set watch in zookeeper
USAGE    : set_watch watch_name watch_path [zk_servers]
*****
COMMAND : del_watch
DESCRIBE : delete watch from zookeeper
USAGE    : del_watch watch_name
```

局限: find\_key 和 find\_keys 两个命令在使用的时候, 输入的 key 不能包含空格。目前仅支持 conf watch。

## 配置中心

作用: 使 twemproxy 在启动的时候可以从配置中心获取配置, 并且当配置中心更改了配置

后，twemproxy 能相应的重新加载配置。

#### 使用方法：

在 twemproxy 编译前 configure 的时候要增加--with-zookeeper 项。

在 nutcracker 的启动项中增加了三个选项： -S(--zk-start) -K(--zk-keep) -Z(--zk-path) -z(--zk-server)。

#### 参数说明：

-S 选项指定 twemproxy 启动时配置从 zookeeper 获取。默认不开启。

-K 选项指定当 zookeeper 改变了该相应配置后，twemproxy 要重新获取配置并加载。默认不开启。

-Z 选项指定 twemproxy 获取和监听 zookeeper 的配置节点名字。默认为 twemproxy。

-z 选项指定 zookeeper 的连接地址。默认为 127.0.0.1:2181。

示例：nutcracker -d -o nutcracker.log -S -K -Z twemproxy123 -z 192.168.0.1:2181, 192.168.0.2:2181

局限：配置内容长度不得超过 5000。

## Replication pool

**作用：**使一个 server pool（slave）成为另外一个 server pool（master）的复制。当存在这种复制关系后，存储命令会同时发送到 master pool 和 slave pool，如果读取命令在 master pool 中 miss，则会穿透到 slave pool 进行查询并返回结果，如果在 slave pool 中 hit 到结果，则可以回写到 master pool 中。

**使用方法：**yaml 配置文件中增加了三个参数：replication\_from, replication\_mode 和 write\_back\_mode。

#### 参数说明：

replication\_from 参数在 slave pool 中配置，指定一个字符串即 master pool 的名字，此时 slave pool 就成为了 master pool 的 replication pool。

replication\_mode 参数在 master pool 中配置，可以指定三个值：0（默认），1 和 2。replication\_mode 作用于存储命令，当 replication\_mode:0，nutcracker 不关心 slave pool 的执行结果，只返回 master pool 的结果给用户；当 replication\_mode:1，如果 master pool 和 slave pool 执行结果不一致时，记录到 log 文件，nutcracker 只返回 master pool 的结果给用户；当 replication\_mode:2，如果 master pool 和 slave pool 执行结果不一致，nutcracker 返回 master pool 和 slave pool 中最坏的结果给用户。

write\_back\_mode 参数在 master pool 中配置，可以指定两个值：0（默认）和 1。write\_back\_mode 作用于读取命令，当 write\_back\_mode:0，穿透以后，如果在 slave pool 中 hit 结果，不回写；当 write\_back\_mode:1，穿透以后，如果在 slave pool 中 hit 结果，回写结果到 master pool 中。

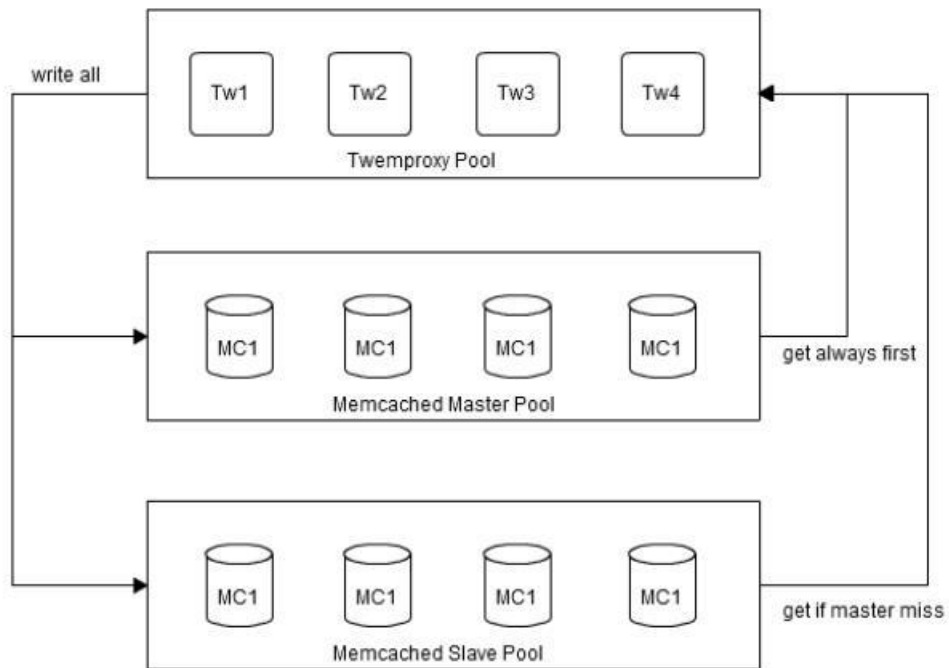
#### 示例：

twem1:

listen: 127.0.0.1:22122  
hash: fnv1a\_64  
hash\_tag: "{}"  
distribution: ketama  
auto\_eject\_hosts: true  
timeout: 400  
redis: false  
replication\_mode: 1  
write\_back\_mode: 1  
server\_connections: 5  
servers:  
- 127.0.0.1:23401:1 server1  
- 127.0.0.1:23401:1 server2  
- 127.0.0.1:23401:1 server3

twem2:

listen: 127.0.0.1:22123  
hash: fnv1a\_64  
hash\_tag: "{}"  
distribution: ketama  
auto\_eject\_hosts: true  
timeout: 400  
redis: false  
server\_connections: 5  
replication\_from: twem1  
servers:  
- 127.0.0.1:23402:1 server1  
- 127.0.0.1:23402:1 server2  
- 127.0.0.1:23402:1 server3



其中 server pool twem2 就是 server pool twem1 的 replication pool。

**局限：**master pool 和 slave pool 仅限在同一个 yml 配置文件。目前仅支持 redis:false。暂不支持一主多从。