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：**

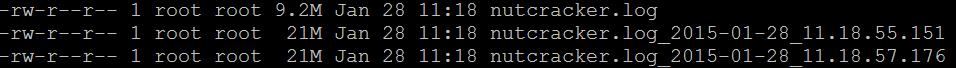
**-R选项用于启用日志切割功能。默认不开启。**

**-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**

****

**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**



**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：yml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salve 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**

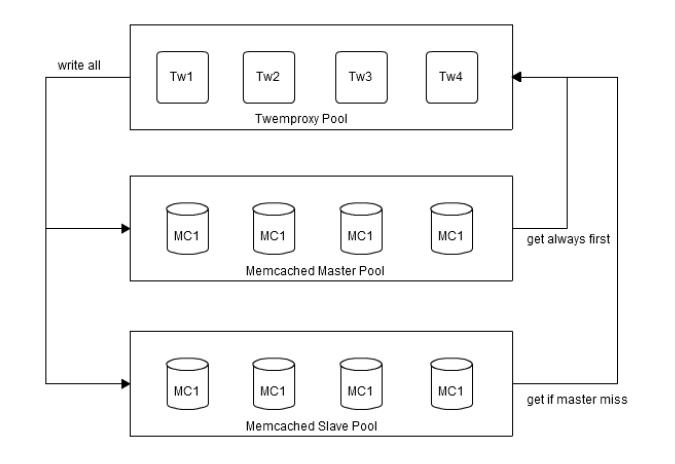
**replication\_from: twem1**

**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。**