Twemproxy是一种代理分片机制，由Twitter开源。

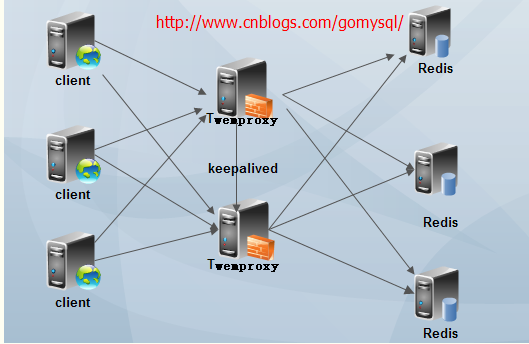
Twemproxy作为代理，可接受来自多个程序的访问，按照路由规则，转发给后台的各个Redis服务器，再原路返回。

该方案很好的解决了单个Redis实例承载能力的问题。当然，Twemproxy本身也是单点，需要用Keepalived做高可用方案。

通过Twemproxy可以使用多台服务器来水平扩张redis服务，可以有效的避免单点故障问题。虽然使用Twemproxy需要更多的硬件资源和在redis性能有一定的损失（twitter测试约20%），但是能够提高整个系统的HA也是相当划算的。

twemproxy不光可以代理redis，还可以代理memcached，官方说明：

通常会结合keepalived来实现Twemproxy的高可用。架构图如下：



上面的架构通常只有一台Twemproxy在工作，另外一台处于备机，当一台挂掉以后，vip自动漂移，备机接替工作。关于keepalived相关的资料可以参考我前面相关文章。

1.编译安装：

autoconf下载地址：[http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz](http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz" \o "[此链接已破解] http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz" \t "_blank)  
twemproxy下载地址：[https://codeload.github.com/twitter/twemproxy/zip/master](https://codeload.github.com/twitter/twemproxy/zip/master" \o "https://codeload.github.com/twitter/twemproxy/zip/master" \t "_blank)  
twemproxy的安装要求autoconf的版本在2.64以上，否则提示"error: Autoconf version 2.64 or higher is required"。

查找旧版本autoconf，并且卸载

rpm -qf /usr/bin/autoconf

rpm -e --nodeps autoconf-2.63

安装最新版本

tar zxvf autoconf-2.69.tar.gz

cd autoconf-2.69

./configure --prefix=/usr

make && make install

编译安装twemproxy

unzip twemproxy-master.zip

cd twemproxy-master

autoreconf -fvi

./configure --prefix=/usr/local/twemproxy

make -j 8

make install

设置环境变量：

echo "PATH=$PATH:/usr/local/twemproxy/sbin/" >> /etc/profile

source /etc/profile

2.创建相关目录（存放配置文件和pid文件）

cd /usr/local/twemproxy

mkdir run conf

3.添加proxy配置文件

vim /usr/local/twemproxy/conf/nutcracker.yml

内容如下（配置文件参考官方）

[复制代码](javascript:void(0);)

复制代码

alpha:

listen: 127.0.0.1:22121

hash: fnv1a\_64

distribution: ketama

auto\_eject\_hosts: true

redis: true

server\_retry\_timeout: 2000

server\_failure\_limit: 1

servers:

- 127.0.0.1:7000:1

- 127.0.0.1:7001:1

- 127.0.0.1:7002:1

- 127.0.0.1:7003:1

- 127.0.0.1:7004:1

- 127.0.0.1:7005:1

复制代码

[复制代码](javascript:void(0);)

我在本地安装了6个redis实例

redis官网安装[Redis安装以及主从实现](http://www.cnblogs.com/gomysql/p/3617238.html" \t "_blank)。

4.启动Twemproxy服务

nutcracker -t 测试配置文件

把conf目录复制到/usr/local/twemproxy/sbin/目录下，进行检测：

[root@redis-server sbin]# ./nutcracker -t

nutcracker: configuration file 'conf/nutcracker.yml' syntax is ok

[root@redis-server sbin]#

启动命令：

nutcracker -d -c /usr/local/twemproxy/conf/nutcracker.yml -p /usr/local/twemproxy/run/redisproxy.pid -o /usr/local/twemproxy/run/redisproxy.log

查看进程，确认启动。

[root@redis-server run]# ps -ef | grep nutcracker | grep -v grep

root 809 1 0 03:09 ? 00:00:00 nutcracker -d -c

5. 简单测试

[root@redis-server ~]# netstat -nltp | grep nutcracker

tcp 0 0 0.0.0.0:22222 0.0.0.0:\* LISTEN 809/nutcracker

tcp 0 0 127.0.0.1:22121 0.0.0.0:\* LISTEN 809/nutcracker

[root@redis-server ~]# redis-cli -p 22121

127.0.0.1:22121> set name yaun

OK

127.0.0.1:22121> get name

"yaun"

127.0.0.1:22121>

**性能测试**

这里使用redis自带的redis-benchmark进行简单的性能测试，测试结果如下:

1. Set测试：
   1. 通过twemproxy测试：

[root@COS1 src]# redis-benchmark -h 10.23.22.240 -p 22121 -c 100 -t set -d 100 -l –q

SET: 38167.94 requests per second

* 1. 直接对后端redis测试：

[root@COS2 ~]# redis-benchmark -h 10.23.22.241 -p 6379 -c 100 -t set -d 100 -l –q

SET: 53191.49 requests per second

1. Get测试：
   1. 通过twemproxy测试：
   2. [root@COS1 src]# redis-benchmark -h 10.23.22.240 -p 22121 -c 100 -t get -d 100 -l -q

GET: 37453.18 requests per second

* 1. 直接对后端redis测试：
  2. [root@COS2 ~]# redis-benchmark -h 10.23.22.241 -p 6379 -c 100 -t get -d 100 -l -q

GET: 62111.80 requests per second

1. 查看键值分布：
2. [root@COS2 ~]# redis-cli info|grep db0
3. db0:keys=51483,expires=0,avg\_ttl=0
4. [root@COS3 ~]# redis-cli info|grep db0

db0:keys=48525,expires=0,avg\_ttl=0

**nutcracker用法与命令选项**

Usage: nutcracker [-?hVdDt] [-v verbosity level] [-o output file]

[-c conf file] [-s stats port] [-a stats addr]

[-i stats interval] [-p pid file] [-m mbuf size]

[复制代码](javascript:void(0);)

复制代码

Options:

-h, –help : 查看帮助文档，显示命令选项

-V, –version : 查看nutcracker版本

-t, –test-conf : 测试配置脚本的正确性

-d, –daemonize : 以守护进程运行

-D, –describe-stats : 打印状态描述

-v, –verbosity=N : 设置日志级别 (default: 5, min: 0, max: 11)

-o, –output=S : 设置日志输出路径，默认为标准错误输出 (default: stderr)

-c, –conf-file=S : 指定配置文件路径 (default: conf/nutcracker.yml)

-s, –stats-port=N : 设置状态监控端口，默认22222 (default: 22222)

-a, –stats-addr=S : 设置状态监控IP，默认0.0.0.0 (default: 0.0.0.0)

-i, –stats-interval=N : 设置状态聚合间隔 (default: 30000 msec)

-p, –pid-file=S : 指定进程pid文件路径，默认关闭 (default: off)

-m, –mbuf-size=N : 设置mbuf块大小，以bytes单位 (default: 16384 bytes)