

- 安装memcached
- 使用memcached
  - 命令
  - java客户端
- 集群
  - twemproxy (nutcracker)
    - 1、概述
    - 2、环境准备(伪集群/真集群都一样)
  - twemproxy 部署
    - 1、安装 依赖
    - 3、安装 twemproxy代理
    - 4、查看 twemproxy帮助
    - 5、准备 twemproxy配置文件
- 检查语法
- 启动
  - 后台启动 /usr/local/twemproxy/sbin/nutcracker -d

## 安装memcached

```
1 yum install libevent-devel
2 wget https://memcached.org/latest
3 mv latest memcached-1.5.12.tar.gz
4 tar -zxf memcached-1.5.12.tar.gz
5 cd memcached-1.5.12
6 ./configure --prefix=/usr/local/memcached
7 make && sudo make install
8
9 -p <num> 监听的TCP端口(默认: 11211)
10 -U <num> 监听的UDP端口(默认: 11211, 0表示不监听)
11 -l <ip_addr> 监听的IP地址。(默认: INADDR_ANY, 所有地址)
12 -d 作为守护进程来运行
13 -u <username> 设定进程所属用户(仅root用户可以使用)
14 -m <num> 所有slab class可用内存的上限(默认: 64MB)
15 -v 提示信息(在事件循环中打印错误/警告信息。)
16 -vv 详细信息(还打印客户端命令/响应)
```

```
17 -vvv 超详细信息（还打印内部状态的变化）
18
19
```

## 使用memcached

- 命令

```
1 # telnet memcached_server_ip port例如:
2 telnet localhost 11211
```

- java客户端

```
1 xmemcached: https://github.com/killme2008/xmemcached
```

## 集群

### twemproxy (nutcracker)

#### 1、概述

twemproxy (nutcracker) 是 Twitter开源的轻量级 memcached / redis 代理服务器，本质就是一个集群管理工具，主要用来弥补 Redis和 Memcached对集群管理的不足，其完成的最大功劳就是通过在后端减少同缓存服务器的连接数从而增加吞吐量

#### 2、环境准备(伪集群/真集群都一样)

准备三个节点：

节点	端口	名字
127.0.0.1	11211	node1
127.0.0.1	11212	node2
127.0.0.1	22121	proxy

备注：这里我们是用的同一台机器，可以根据自己的情况配置相应 ip 和 端口

## twemproxy 部署

### 1、安装 依赖

```
1 yum install autoconf
2 yum install automake
3 yum install
```

### 3、安装 twemproxy代理

```
1 wget https://github.com/twitter/twemproxy/archive/master.zip
2 unzip master.zip
3 cd twemproxy-master
4 autoreconf -fvi
5 ./configure --prefix=/usr/local/twemproxy
6 make
7 make install
```

### 4、查看 twemproxy帮助

```
/usr/local/twemproxy/sbin/nutcracker -h
```

### 5、准备 twemproxy配置文件

...

```
mkdir /usr/local/twemproxy/conf/
```

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

```
1 - 修改配置文件
```

```
memcached:
listen: 127.0.0.1:22121
hash: fnv1a_64
distribution: ketama
timeout: 400
backlog: 1024
preconnect: true
auto_eject_hosts: true
server_retry_timeout: 30000
server_failure_limit: 3
servers:

  • 127.0.0.1:11211:1
  • 127.0.0.1:11212:1
```

```
1 ### 6、启动twemproxy
```

## 检查语法

```
/usr/local/twemproxy/sbin/nutcracker -t
```

## 启动

```
/usr/local/twemproxy/sbin/nutcracker
```

## 后台启动 /usr/local/twemproxy/sbin/nutcracker -d

```
1 ### 7、数据读/写测试
2
3 - ##### 首先通过 twemproxy代理来写缓存
```

```
[root@localhost conf]# telnet localhost 22121
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
set key1 0 0 1
1
STORED
set key2 0 0 1
2
STORED
set key3 0 0 1
3
STORED
set key4 0 0 1
4
STORED
set key5 0 0 1
5
STORED
set key6 0 0 1
6
STORED
```

```
1 - ##### 查看发现所有缓存都写到了 memcached2中
```

```
[root@localhost ~]# ps -aux | grep mem
root 634 0.0 0.0 326588 1960 ? Ssl 15:58 0:00 memcached -u root -p 11211 -m 64m
-d
root 704 0.0 0.0 112676 984 pts/0 S+ 16:01 0:00 grep --color=auto mem
[root@localhost ~]# kill -9 634
```

```
1 - ##### 继续通过 twemproxy代理来写缓存
```

```
[root@localhost conf]# telnet localhost 22121
Trying ::1...
telnet: connect to address ::1: Connection refused
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^'.
set key9 0 0 1
9
STORED
[root@localhost conf]#
```

```
1 - ##### 此时去memcached1查看:
```

```
[root@localhost ~]# telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^'.
get key9
VALUE key9 0 1
9
END
```

```
1 - ##### 重新启动 memcached2
```

```
[root@localhost ~]# telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^'.
get key10
```

```
VALUE key10 0 1
```

```
x
```

```
END
```

```
get key11
```

```
VALUE key11 0 1
```

```
y
```

```
END
```

```
...
```

==总结：上面这个实验过程可以看出，一台 memcached实例挂掉后，twemproxy 能自动移除之；而恢复后，twemproxy 能够自动识别并重新加入到 memcached 组中重新使用==

<https://blog.csdn.net/houjixin/article/details/52101001>

可以选择的key值的hash算法：one\_at\_a\_time、md5、crc16、crc32、crc32a、fnv1\_64、fnv1a\_64、fnv1\_32、fnv1a\_32、hsieh、murmur、jenkins，如果没有选择，默认是fnv1a\_64。