



链滴

# centos7.4 memcache 高可用方案搭建 (3台)

作者: [centrexzj](#)

原文链接: <https://ld246.com/article/1536646434510>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# centos7.4 memcache 高可用方案搭建

## 1. 方案选型

缓存方案的基本要求：避免单点故障；较好的性能和稳定性；便于运维管理。

常见方案如下：

- A. 客户端直接访问多个memcache 实例

优点：简单，未引入新的节点；

缺点：维护不方便，未实现集中管理；性能不满足，实例宕机后不能自动踢出（hash到该实例的请求要等到超时才能转到其他正常实例）。

- B. magent代理

优点：简单，满足缓存对代理的大部分要求；

缺点：无成熟案例；性能不满足，实例宕机后不能自动踢出。

- C. moxi代理

优点：功能丰富，提供本地缓存、Memcache实例数据复制等。

缺点：无成熟案例；代码很久没更新。

- D. twemproxy代理

优点：twitter的成熟案例。

缺点：主要配合是twemproxy+redis，与memcache配合使用的案例较少；不支持缓存的复制功能。

1. 下载地址：<https://github.com/twitter/twemproxy>

2. docker方案下载地址：<https://github.com/elegos/docker-twemproxy>

- E. mcrouter

优点：facebook的成熟案例；功能强大，支持Memcache实例分组、实例复制功能，实例宕机后可动踢出。

缺点：一般用于ubuntu平台，在centos上安装较复杂。

1. 下载地址：<https://github.com/facebook/mcrouter>

2. docker方案下载地址：<https://github.com/kakerukaeru/mcrouter-docker>

## 2. memcache安装

Memcache用到了libevent这个库用于Socket的处理，所以还需要安装libevent，libevent的最新版是libevent-1.3。（如果你的系统已经安装了libevent，可以不用安装）

```
[root@memcache1 ~]# yum install libevent  
yum -y install libevent-devel
```

下载memcached

```
wget http://www.memcached.org/files/memcached-1.5.7.tar.gz
```

```
tar zxvf memcached-1.5.7.tar.gz
cd memcached-1.5.7
./configure
make -j8
make install
```

### 3. 启动memcache

创建memcached启动用户

```
useradd memcache
passwd memcache
```

启动memcache

```
chmod 777 -R /var/run/
memcached -m 32m -p 11211 -d -u memcache -P /var/run/memcached.pid -c 256 -vv
```

参数说明:

- d选项是作为守护进程在后台运行
- m是分配给Memcache使用的内存数量, 单位是MB, 我这里是10MB,
- u是运行Memcache的用户, 我这里是root,
- l是监听的服务器IP地址, 如果有多个地址的话
- p是设置Memcache监听的端口, 我这里设置了12000, 最好是1024以上的端口,
- c选项是最大运行的并发连接数, 默认是1024, 我这里设置了256, 按照你服务器的负载量来设定,
- P是设置保存Memcache的pid文件
- vv是以very vrebosc模式启动, 将调试信息和错误输出到控制台

```
[root@memcache1 memcached-1.5.7]# netstat -unlpt | grep 11211
tcp      0      0 0.0.0.0:11211      0.0.0.0:*        LISTEN   10776/memcached
tcp6     0      0 :::11211          :::*              LISTEN   10776/memcached
```

关闭memcached的命令为:

```
kill `cat /var/run/memcached.pid`
```

### 设置memcache自动启动

- > vim /etc/init.d/memcached # 新建memcached文件保存
- > cp /usr/local/src/memcached-1.5.7/scripts/memcached.sysv /etc/init.d/memcached #需要脚本中的memcached目录改为自己的memcached目录即可
- > chmod 755 /etc/init.d/memcached #修改文件权限
- > chkconfig --add memcached #将memcache加入系统服务
- > chkconfig memcached on
- > service memcached start #启动, 也可以用/etc/init.d/memcached start

memcached内容

```
#!/bin/sh
#
# chkconfig: - 55 45
# description: The memcached daemon is a network memory cache service.
```

```

# processname: memcached
# config: /etc/sysconfig/memcached

# Source function library.
./etc/rc.d/init.d/functions

PORT=11211
USER=memcache
MAXCONN=10240
CACHESIZE=64
OPTIONS=""

if [ -f /etc/sysconfig/memcached ];then
    ./etc/sysconfig/memcached
fi

# Check that networking is up.
if [ "$NETWORKING" = "no" ]
then
    exit 0
fi

RETVAL=0
prog="memcached"

start () {
    echo -n $"Starting $prog: "
    # insure that /var/run/memcached has proper permissions
    chown $USER /var/run/memcached
    daemon /usr/local/src/memcached-1.5.7/memcached -d -p $PORT -u $USER -m $CACHES
    ZE -c $MAXCONN -P /var/run/memcached/memcached.pid $OPTIONS
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/memcached
}
stop () {
    echo -n $"Stopping $prog: "
    killproc memcached
    RETVAL=$?
    echo
    if [ $RETVAL -eq 0 ]; then
        rm -f /var/lock/subsys/memcached
        rm -f /var/run/memcached/memcached.pid
    fi
}

restart () {
    stop
    start
}

# See how we were called.
case "$1" in

```

```

start)
    start
    ;;
stop)
    stop
    ;;
status)
    status memcached
    ;;
restart|reload)
    restart
    ;;
condrestart)
    [ -f /var/lock/subsys/memcached ] && restart || :
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart|reload|condrestart}"
    exit 1
esac

exit $?

```

然后把/etc/init.d/memcached复制到另外2台电脑

```

> chmod 755 /etc/init.d/memcached #修改文件权限
> chkconfig --add memcached #将memcache加入系统服务
> chkconfig memcached on

```

## 安装twemproxy

```

sbin/nutcracker -d -c /usr/local/nutcracker/conf/nutcracker.yml -p /usr/local/nutcracker/run/
emproxy.pid -o /usr/local/nutcracker/run/memproxy.log

```

```

nc 192.168.31.226 22222 | python -mjson.tool

```

```

curl http://192.168.31.226:22222

```

- 自动启动

```

vim vim /etc/rc.local

```

```

/usr/local/nutcracker/sbin/nutcracker -d -c /usr/local/nutcracker/conf/nutcracker.yml -p /usr/
ocal/nutcracker/run/memproxy.pid -o /usr/local/nutcracker/run/memproxy.log

```

```

chmod +x /etc/rc.d/rc.local

```

[memcached下载地址](#)