一、Redis介绍

Redis是当前比较热门的NOSQL系统之一，它是一个key-value存储系统。和Memcache类似，但很大程度补偿了Memcache的不足，它支持存储的value类型相对更多，包括string、list、set、zset和hash。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作。在此基础上，Redis支持各种不同方式的排序。

和Memcache一样，Redis数据都是缓存在计算机内存中，不同的是，Memcache只能将数据缓存到内存中，无法自动定期写入硬盘，这就表示，一断电或重启，内存清空，数据丢失。所以Memcache的应用场景适用于缓存无需持久化的数据。而Redis不同的是它会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，实现数据的持久化。

二、Redis的安装

下面介绍在Linux环境下，Redis的安装与部署

1、首先上官网下载Redis 压缩包，地址：http://redis.io/download 下载稳定版3.0.7即可。

2、通过远程管理工具，将压缩包拷贝到Linux服务器中，执行解压操作

```
login as: root
root@192.168.26.128's password:
[root@localhost ~]# cd /lamp
[root@localhost lamp]# tar zxvf redis-3.0.7.tar.gz
redis-3.0.7/
redis-3.0.7/.gitignore
redis-3.0.7/00-RELEASENOTES
redis-3.0.7/BUGS
redis-3.0.7/CONTRIBUTING
redis-3.0.7/COPYING
redis-3.0.7/INSTALL
redis-3.0.7/MANIFESTO
redis-3.0.7/Makefile
redis-3.0.7/README
redis-3.0.7/deps/
redis-3.0.7/deps/Makefile
redis-3.0.7/deps/hiredis/
redis-3.0.7/deps/hiredis/.gitignore
```


3、执行make 对Redis解压后文件进行编译

```
[root@localhost lamp]# cd redis-3.0.7
[root@localhost redis-3.0.7]# make
cd src && make all
make[1]: Entering directory `/lamp/redis-3.0.7/src'
rm -rf redis-server redis-sentinel redis-cli redis-benchmark redis-check-dump re
dis-check-aof *.o *.gcda *.gcno *.gcov redis.info lcov-html
(cd ../deps && make distclean)
make[2]: Entering directory `/lamp/redis-3.0.7/deps'
(cd hiredis && make clean) > /dev/null || true
(cd linenoise && make clean) > /dev/null || true
(cd lua && make clean) > /dev/null || true
(cd jemalloc && [ -f Makefile ] && make distclean) > /dev/null || true
(rm -f .make-*)
make[2]: Leaving directory `/lamp/redis-3.0.7/deps'
(rm -f .make-*)
echo STD=-std=c99 -pedantic >> .make-settings
echo WARN=-Wall -W >> .make-settings
echo OPT=-O2 >> .make-settings
echo MALLOC=jemalloc >> .make-settings
echo CFLAGS= >> .make-settings
echo LDFLAGS= >> .make-settings
echo REDIS_CFLAGS= >> .make-settings
echo REDIS_LDFLAGS= >> .make-settings
echo PREV_FINAL_CFLAGS=-std=c99 -pedantic -Wall -W -O2 -g -ggdb   -I../deps/hire
```


编译完成之后，可以看到解压文件redis-3.0.7 中会有对应的src、conf等文件夹，这和windows下安装解压的文件一样，大部分安装包都会有对应的类文件、配置文件和一些命令文件。

```
Hint: It's a good idea to run 'make test' ;)

make[1]: Leaving directory `/lamp/redis-3.0.7/src'
[root@localhost redis-3.0.7]# ll
total 152
-rw-rw-r--.  1 root root 36761 Jan 25 06:57 00-RELEASENOTES
-rw-rw-r--.  1 root root    53 Jan 25 06:57 BUGS
-rw-rw-r--.  1 root root  1805 Jan 25 06:57 CONTRIBUTING
-rw-rw-r--.  1 root root  1487 Jan 25 06:57 COPYING
drwxrwxr-x.  6 root root  4096 Jan 31 02:50 deps
-rw-rw-r--.  1 root root    11 Jan 25 06:57 INSTALL
-rw-rw-r--.  1 root root   151 Jan 25 06:57 Makefile
-rw-rw-r--.  1 root root  4223 Jan 25 06:57 MANIFESTO
-rw-rw-r--.  1 root root  5201 Jan 25 06:57 README
-rw-rw-r--.  1 root root 41560 Jan 25 06:57 redis.conf
-rwxrwxr-x.  1 root root   271 Jan 25 06:57 runtest
-rwxrwxr-x.  1 root root   280 Jan 25 06:57 runtest-cluster
-rwxrwxr-x.  1 root root   281 Jan 25 06:57 runtest-sentinel
-rw-rw-r--.  1 root root  7109 Jan 25 06:57 sentinel.conf
drwxrwxr-x.  2 root root  4096 Jan 31 02:55 src
drwxrwxr-x. 10 root root  4096 Jan 25 06:57 tests
drwxrwxr-x.  5 root root  4096 Jan 25 06:57 utils
[root@localhost redis-3.0.7]# cd src
[root@localhost src]# make install
```

4、编译成功后，进入src文件夹，执行make install进行Redis安装

5、安装完成，界面如下

```
[root@localhost redis-3.0.7]# cd src
[root@localhost src]# make install

Hint: It's a good idea to run 'make test' ;)

    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
[root@localhost src]# ll
total 24136
-rw-rw-r--. 1 root root    9927 Jan 25 06:57 adlist.c
-rw-rw-r--. 1 root root    3451 Jan 25 06:57 adlist.h
-rw-r--r--. 1 root root   10124 Jan 31 02:53 adlist.o
-rw-rw-r--. 1 root root   15854 Jan 25 06:57 ae.c
-rw-rw-r--. 1 root root    4876 Jan 25 06:57 ae_epoll.c
-rw-rw-r--. 1 root root   10939 Jan 25 06:57 ae_evport.c
-rw-rw-r--. 1 root root    4631 Jan 25 06:57 ae.h
-rw-rw-r--. 1 root root    4567 Jan 25 06:57 ae_kqueue.c
-rw-r--r--. 1 root root   21128 Jan 31 02:53 ae.o
-rw-rw-r--. 1 root root    3804 Jan 25 06:57 ae_select.c
-rw-rw-r--. 1 root root   19743 Jan 25 06:57 anet.c
-rw-rw-r--. 1 root root    3361 Jan 25 06:57 anet.h
-rw-r--r--. 1 root root   29236 Jan 31 02:53 anet.o
-rw-rw-r--. 1 root root   59937 Jan 25 06:57 aof.c
-rw-r--r--. 1 root root   76308 Jan 31 02:54 aof.o
-rw-rw-r--. 1 root root    2833 Jan 25 06:57 asciilogo.h
-rw-rw-r--. 1 root root    8951 Jan 25 06:57 bio.c
-rw-rw-r--. 1 root root    2091 Jan 25 06:57 bio.h
-rw-r--r--. 1 root root   11948 Jan 31 02:55 bio.o
```


三、Redis的部署

安装成功后，下面对Redis 进行部署

1、首先为了方便管理，将Redis文件中的conf配置文件和常用命令移动到统一文件中

a)创建bin和redis.conf文件

复制代码代码如下:

```
mkdir -p/usr/local/redis/bin
mkdir -p/usr/local/redis/ect
```

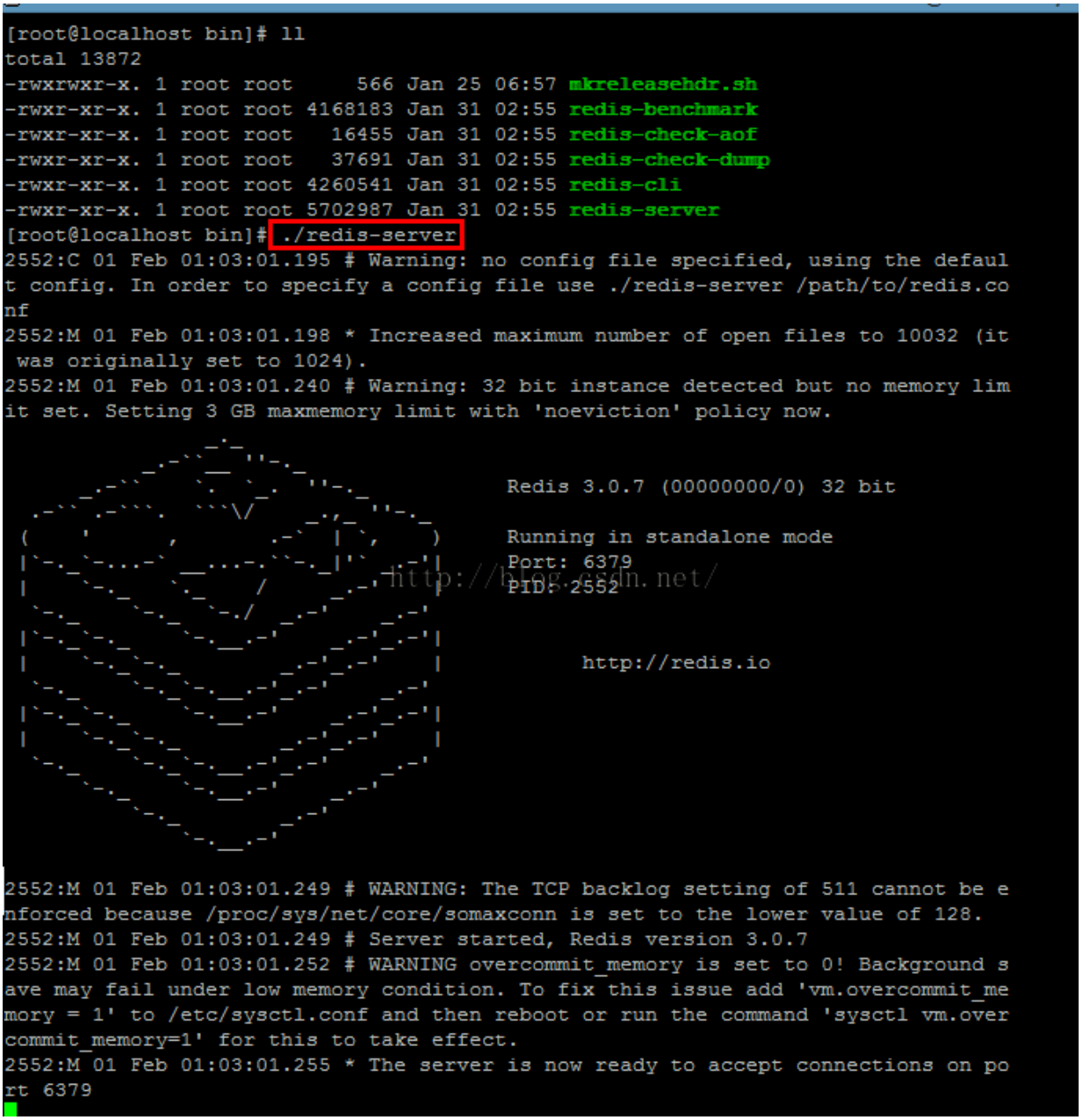
b)执行Linux文件移动命令：

复制代码代码如下:

```
mv /lamp/redis-3.0.7/redis.conf /usr/local/redis/etc
cd /lamp/redis-3.0.7/src
```

mv mkreleasdhdr.sh redis-benchmark redis-check-aof redis-check-dump redis-cli redis-server /usr/local/redis/bin

2、执行Redis-server 命令，启动Redis 服务



注意：这里直接执行Redis-server 启动的Redis服务，是在前台直接运行的(效果如上图)，也就是说，执行完该命令后，如果Lunix关闭当前会话，则Redis服务也随即关闭。正常情况下，启动Redis服务需要从后台启动，并且指定启动配置文件。

3、后台启动redis服务

a)首先编辑conf文件，将daemonize属性改为yes（表明需要在后台运行）

cd etc/

Vi redis.conf

b)再次启动redis服务，并指定启动服务配置文件

redis-server /usr/local/redis/etc/redis.conf

```
login as: root
root@192.168.26.128's password:
Last login: Mon Feb  1 01:25:00 2016 from 192.168.26.1
[root@localhost ~]# cd /usr/local/redis/etc
[root@localhost etc]# ll
total 44
-rw-rw-r--. 1 root root 41560 Jan 25 06:57 redis.conf
[root@localhost etc]# vi redis.conf
[root@localhost etc]# vi redis.conf
[root@localhost etc]# ps -ef | grep redis
root      2552     1  0 01:03 ?        00:00:04 ./redis-server *:6379
root      2649  2627  0 01:39 pts/1    00:00:00 grep redis
[root@localhost etc]# netstat -tunpl |grep 6379
tcp        0      0 0.0.0.0:6379                0.0.0.0:*                   LISTEN      2552/./redis-server
tcp        0      0 :::6379                     :::*                        LISTEN      2552/./redis-server
[root@localhost etc]# cd /bin
[root@localhost bin]# redis-cli
127.0.0.1:6379>
127.0.0.1:6379>                                  http://blog.csdn.net/
127.0.0.1:6379> exit
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]# pkill redis-server
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]# !net
netstat -tunpl |grep 6379
[root@localhost bin]# ./redis-cli
-bash: ./redis-cli: No such file or directory
[root@localhost bin]# redis-cli
Could not connect to Redis at 127.0.0.1:6379: Connection refused
not connected>
not connected>
not connected> exit
```

4、服务端启动成功后，执行redis-cli启动Redis 客户端，查看端口号。

```
not connected> exit
[root@localhost bin]# redis-server /usr/local/redis/etc/redis.conf
[root@localhost bin]#
[root@localhost bin]# redis-cli
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> exit
[root@localhost bin]# netstat -tunpl|grep 6379
tcp        0      0 0.0.0.0:6379                0.0.0.0:*                   LISTEN      2675/redis-server *
tcp        0      0 :::6379                     :::*                        LISTEN      2675/redis-server *
[root@localhost bin]#                  http://blog.csdn.net/
[root@localhost bin]#
[root@localhost bin]# redis-cli shutdown
[root@localhost bin]#
[root@localhost bin]#
[root@localhost bin]# netstat -tunpl|grep 6379
[root@localhost bin]#
[root@localhost bin]#
```

vim /etc/init.d/redis

#!/bin/sh

#

redis   Startup script for Redis Server

#

chkconfig: - 80 12

description: Redis is an open source, advanced key-value store.

#

processname: redis-server

config: /etc/redis.conf

```
# pidfile: /var/run/redis.pid
source /etc/init.d/functions
BIN="/usr/local/redis/bin"
CONFIG="/usr/local/redis/redis.conf"
PIDFILE="/var/run/redis.pid"
### Read configuration
[ -r "$SYSCONFIG" ] && source "$SYSCONFIG"
RETVAL=0
prog="redis-server"
desc="Redis Server"
start() {
        if [ -e $PIDFILE ];then
            echo "$desc already running...."
            exit 1
        fi
        echo -n $"Starting $desc: "
        daemon $BIN/$prog $CONFIG
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$prog
        return $RETVAL
}
stop() {
        echo -n $"Stop $desc: "
        killproc $prog
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/$prog $PIDFILE
        return $RETVAL
}
restart() {
        stop
        start
}
case "$1" in
```

```
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  condrestart)
        [ -e /var/lock/subsys/$prog ] && restart
        RETVAL=$?
        ;;
  status)
        status $prog
        RETVAL=$?
        ;;
   *)
        echo $"Usage: $0 {start|stop|restart|condrestart|status}"
        RETVAL=1
esac
exit $RETVAL
```

```
chmod +x /etc/init.d/redis
service redis start
service redis stop
chkconfig --add redis
```

修改profile文件：

```
# vi /etc/profile
```

在最后行添加:

```
export PATH="$PATH:/usr/local/redis/bin"
```

然后马上应用这个文件：

```
# . /etc/profile
```

配置下面的内核参数，否则Redis脚本在重启或停止redis时，将会报错，并且不能自动在停止服务前同步数据到磁盘上/etc/sysctl.conf加上

#vim /etc/sysctl.conf

vm.overcommit_memory = 1

#sysctl -p

四、总结Linux 、Redis 操作常用命令

Linux :

cd /usr 从子文件夹进入上级文件夹usr

cd local 从父到子

mv /A /B 将文件A移动到B

vi usr/local/redis/redis.conf 编辑redis.conf 文件

:wq 保存修改，并退出

Redis :

Redis-server /usr..../redis.conf 启动redis服务，并指定配置文件

Redis-cli 启动redis 客户端

Pkill redis-server 关闭redis服务

Redis-cli shutdown 关闭redis客户端

Netstat -tunpl|grep 6379 查看redis 默认端口号6379占用情况

4. Redis的配置

4.1. Redis默认不是以守护进程的方式运行，可以通过该配置项修改，使用yes启用守护进程

daemonize no

4.2. 当Redis以守护进程方式运行时，Redis默认会把pid写入/var/run/redis.pid文件，可以通过pidfile指定

pidfile /var/run/redis.pid

4.3. 指定Redis监听端口，默认端口为6379，作者在自己的一篇博文中解释了为什么选用6379作为默认端口，因为6379在手机按键上MERZ对应的号码，而MERZ取自意大利歌女Alessia Merz的名字

port 6379

4.4. 绑定的主机地址

bind 127.0.0.1

4.5.当 客户端闲置多长时间后关闭连接，如果指定为0，表示关闭该功能

timeout 300

4.6. 指定日志记录级别，Redis总共支持四个级别：debug、verbose、notice、warning，默认为verbose

loglevel verbose

4.7. 日志记录方式，默认为标准输出，如果配置Redis为守护进程方式运行，而这里又配置为日志记录方式为标准输出，则日志将会发送给/dev/null

logfile stdout

4.8. 设置数据库的数量，默认数据库为0，可以使用SELECT <dbid>命令在连接上指定数据库id

databases 16

4.9. 指定在多长时间内，有多少次更新操作，就将数据同步到数据文件，可以多个条件配合

save <seconds> <changes>

Redis默认配置文件中提供了三个条件：

save 900 1

save 300 10

save 60 10000

分别表示900秒（15分钟）内有1个更改，300秒（5分钟）内有10个更改以及60秒内有10000个更改。

4.10. 指定存储至本地数据库时是否压缩数据，默认为yes，Redis采用LZF压缩，如果为了节省CPU时间，可以关闭该选项，但会导致数据库文件变的巨大

rdbcompression yes

4.11. 指定本地数据库文件名，默认值为dump.rdb

dbfilename dump.rdb

4.12. 指定本地数据库存放目录

dir ./

4.13. 设置当本机为slav服务时，设置master服务的IP地址及端口，在Redis启动时，它会自动从master进行数据同步

slaveof <masterip> <masterport>

4.14. 当master服务设置了密码保护时，slav服务连接master的密码

masterauth <master-password>

4.15. 设置Redis连接密码，如果配置了连接密码，客户端在连接Redis时需要通过AUTH <password>命令提供密码，默认关闭

requirepass foobared

4.16. 设置同一时间最大客户端连接数，默认无限制，Redis可以同时打开的客户端连接数为Redis进程可以打开的最大文件描述符数，如果设置 maxclients 0，表示不作限制。当客户

端连接数到达限制时，Redis会关闭新的连接并向客户端返回max number of clients reached错误信息

maxclients 128

4.17. 指定Redis最大内存限制，Redis在启动时会把数据加载到内存中，达到最大内存后，Redis会先尝试清除已到期或即将到期的Key，当此方法处理 后，仍然到达最大内存设置，将无法再进行写入操作，但仍然可以进行读取操作。Redis新的vm机制，会把Key存放内存，Value会存放在swap区

maxmemory <bytes>

4.18. 指定是否在每次更新操作后进行日志记录，Redis在默认情况下是异步的把数据写入磁盘，如果不开启，可能会在断电时导致一段时间内的数据丢失。因为 redis本身同步数据文件是按上面save条件来同步的，所以有的数据会在一段时间内只存在于内存中。默认为no

appendonly no

4.19. 指定更新日志文件名，默认为appendonly.aof

appendfilename appendonly.aof

4.20. 指定更新日志条件，共有3个可选值：

no：表示等**操作系统**进行数据缓存同步到磁盘（快）

always：表示每次更新操作后手动调用fsync()将数据写到磁盘（慢，安全）

everysec：表示每秒同步一次（折衷，默认值）

appendfsync everysec

4.21. 指定是否启用虚拟内存机制，默认值为no，简单的介绍一下，VM机制将数据分页存放，由Redis将访问量较少的页即冷数据swap到磁盘上，访问多的页面由磁盘自动换出到内存中（在后面的文章我会仔细分析Redis的VM机制）

vm-enabled no

4.22. 虚拟内存文件路径，默认值为/tmp/redis.swap，不可多个Redis实例共享

vm-swap-file /tmp/redis.swap

4.23. 将所有大于vm-max-memory的数据存入虚拟内存,无论vm-max-memory设置多小,所有索引数据都是内存存储的(Redis的索引数据 就是keys),也就是说,当vm-max-memory设置为0的时候,其实是所有value都存在于磁盘。默认值为0

vm-max-memory 0

4.24. Redis swap文件分成了很多的page，一个对象可以保存在多个page上面，但一个page上不能被多个对象共享，vm-page-size是要根据存储的 数据大小来设定的，作者建议如果存储很多小对象，page大小最好设置为32或者64bytes；如果存储很大大对象，则可以使用更大的page，如果不 确定，就使用默认值

vm-page-size 32

4.25. 设置swap文件中的page数量，由于页表（一种表示页面空闲或使用的bitmap）是在放在内存中的，，在磁盘上每8个pages将消耗1byte的内存。

vm-pages 134217728

4.26. 设置访问swap文件的线程数,最好不要超过机器的核数,如果设置为0,那么所有对swap文件的操作都是串行的，可能会造成比较长时间的延迟。默认值为4

vm-max-threads 4

4.27. 设置在向客户端应答时，是否把较小的包合并为一个包发送，默认为开启

glueoutputbuf yes

4.28. 指定在超过一定的数量或者最大的元素超过某一临界值时，采用一种特殊的哈希**算法**

hash-max-zipmap-entries 64

hash-max-zipmap-value 512

4.29. 指定是否激活重置哈希，默认为开启（后面在介绍Redis的哈希算法时具体介绍）

activerehashing yes

4.30. 指定包含其它的配置文件，可以在同一主机上多个Redis实例之间使用同一份配置文件，而同时各个实例又拥有自己的特定配置文件

include /path/to/local.conf

以上部分来自网络博客，本人做了一些总结