Redis

# 简介

Redis是一个开源的使用ANSI [C语言](http://baike.baidu.com/view/1219.htm)编写、支持网络、可基于内存亦可持久化的日志型、Key-Value[数据库](http://baike.baidu.com/view/1088.htm)，并提供多种语言的API。从2010年3月15日起，Redis的开发工作由VMware主持。

Redis是一个key-value[存储系统](http://baike.baidu.com/view/51839.htm)。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list([链表](http://baike.baidu.com/view/549479.htm))、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些[数据类型](http://baike.baidu.com/view/675645.htm)都 支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排 序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis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，并且在此基础上实现了master-slave(主从)同步。

Redis 是一个高性能的key-value数据库。 redis的出现，很大程度补偿了[memcached](http://baike.baidu.com/view/794242.htm)这类key/value存储的不足，在部 分场合可以对关系数据库起到很好的补充作用。它提供了Python，Ruby，Erlang，PHP客户端，使用很方便。

# 安装

**获取源码、解压、进入源码目录**

**Wget**

**Tar xvf redis-..tar.gz**

**Cd redis**

**Make**

make命令执行完成后，会在当前目录下生成本个[可执行文件](http://baike.baidu.com/view/159830.htm)，分别是redis-server、redis-cli、redis-benchmark、redis-stat，它们的作用如下：

redis-server：Redis服务器的daemon启动程序

redis-cli：Redis命令行操作工具。当然，你也可以用telnet根据其纯文本协议来操作

redis-benchmark：Redis[性能测试](http://baike.baidu.com/view/106187.htm)工具，测试Redis在你的系统及你的配置下的读写性能

redis-stat：Redis状态检测工具，可以检测Redis当前状态参数及延迟状况。

**建立Redis目录**

这个过程不是必须的，只是为了将Redis相关的[资源统一管理](http://baike.baidu.com/view/3876602.htm)而进行的操作。

执行以下命令建立相关目录并拷贝相关文件至目录中：

mkdir -p /usr/local/redis/bin

mkdir -p /usr/local/redis/etc

mkdir -p /usr/local/redis/var

cp redis-server redis-cli redis-benchmark redis-stat /usr/local/redis/bin/

cp redis.conf /usr/local/redis/etc/

**配置参数**

在我们成功安装Redis后，我们直接执行redis-server即可运行Redis，此时它是按照默认配置来运行的（默认配置甚至不是[后台](http://baike.baidu.com/view/179243.htm)运 行）。我们希望Redis按我们的要求运行，则我们需要修改配置文件，Redis的配置文件就是我们上面第二个cp操作的redis.conf文件，它被 我们拷贝到了/usr/local/redis/etc/目录下。修改它就可以配置我们的server了。如何修改？下面是redis.conf的主要配 置参数的意义：

daemonize：是否以[后台](http://baike.baidu.com/view/179243.htm)daemon方式运行

pidfile：pid文件位置

port：监听的端口号

timeout：请求超时时间

loglevel：log信息级别

logfile：log文件位置

databases：开启数据库的数量

save \* \*：保存[快照](http://baike.baidu.com/view/16510.htm)的频率，第一个\*表示多长时间，第三个\*表示执行多少次写操作。在一定时间内执行一定数量的写操作时，自动保存[快照](http://baike.baidu.com/view/16510.htm)。可设置多个条件。

rdbcompression：是否使用压缩

dbfilename：数据[快照](http://baike.baidu.com/view/16510.htm)文件名（只是文件名，不包括目录）

dir：数据[快照](http://baike.baidu.com/view/16510.htm)的保存目录（这个是目录）

appendonly：是否开启appendonlylog，开启的话每次写操作会记一条log，这会提高数据抗风险能力，但影响效率。

appendfsync：appendonlylog如何同步到磁盘（三个选项，分别是每次写都强制调用fsync、每秒启用一次fsync、不调用fsync等待系统自己同步）

下面是一个略做修改后的配置文件内容：

daemonize yes

pidfile /usr/local/redis/var/redis.pid

port 6379

timeout 300

loglevel debug

logfile /usr/local/redis/var/redis.log

databases 16

save 900 1

save 300 10

save 60 10000

rdbcompression yes

dbfilename dump.rdb

dir /usr/local/redis/var/

appendonly no

appendfsync always

glueoutputbuf yes

shareobjects no

shareobjectspoolsize 1024

将上面内容写为redis.conf并保存到/usr/local/redis/etc/目录下

然后在命令行执行：

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

即可在[后台](http://baike.baidu.com/view/179243.htm)启动redis服务，这时你通过

telnet [127.0.0.1](http://baike.baidu.com/view/971216.htm)6379

即可连接到你的redis服务

# 数据结构

redis提供五种数据类型：[string](http://baike.baidu.com/view/1005388.htm)，hash，list，set及zset(sorted set)。

## string（字符串）

string是最简单的类型，你可以理解成与Memcached一模一样的类型，一个key对应一个value，其上支持的操作与Memcached的操作类似。但它的功能更丰富。

redis采用结构sdshdr和sds封装了字符串，字符串相关的操作实现在[源文件](http://baike.baidu.com/view/385166.htm)sds.h/sds.c中。sdshdr

[数据结构](http://baike.baidu.com/view/9900.htm)定义如下：

typedef char \*sds;

struct sdshdr {

long len;

long free;

char buf[];

};

## list(双向链表)

list是一个链表结构，主要功能是push、pop、获取一个范围的所有值等等。操作中key理解为链表的名字。

对list的定义和实现在源文件adlist.h/adlist.c，相关的[数据结构](http://baike.baidu.com/view/9900.htm)定义如下：

// list[迭代器](http://baike.baidu.com/view/925158.htm)

typedef struct listIter {

listNode \*next;

int direction;

} listIter;

// list[数据结构](http://baike.baidu.com/view/9900.htm)

typedef struct list {

listNode \*head;

listNode \*tail;

void \*(\*dup)(void \*ptr);

void (\*free)(void \*ptr);

int (\*match)(void \*ptr, void \*key);

unsigned int len;

listIter iter;

} list;

# 常用命令

就DB来说，Redis成绩已经很惊人了，且不说[memcachedb](http://baike.baidu.com/view/3400999.htm)和tokyocabinet之流，就说原版的memcached，速度似乎也只能达到这个级别。Redis根本是使用内存[存储](http://baike.baidu.com/view/87682.htm)，持久化的关键是这三条指令：SAVE BGSAVE LASTSAVE …