# 一 简介

## 1 概述

redis是基于键值对的nosql缓存存储系统，可以胜任消息队列、任务队列。

redis以字典结构存储数据，并允许其他应用通过TCP协议读取字典中的内容。redis的键值除了可以是字符串，还可以是其他数据类型。与Mysql不同的是，数据在redis中的存储形式和其他程序、语言非常类似。

redis的数据存储在内存中，所以redis性能极高，可以1秒内读写超过10万个键值，但是随之而来的问题是内存中的数据在程序退出后会丢失，redis也因此提供了持久化支持。

redis为每个键设置了存储时间，到期自动删除，所以redis非常胜任缓存系统。

## 2 优势

性能极高：每秒读速度达11万次，写速度8万1千次；

数据类型丰富：支持二进制的Strings,Lists,Hashes,Sets等；

原子操作：redis的操作都是原子性的，同时支持对几个操作全并后的原子性执行；

特性丰富：支持发布订阅、通知、key过期、持久化等特性；

## 3 应用场景

做缓存，消息队列，session共享、购物车

# 二 部署

## 1 官网快速示例

wget http://download.redis.io/releases/redis-4.0.10.tar.gz

tar xzf redis-4.0.10.tar.gz

cd redis-4.0.10

make

打开服务端窗口： src/redis-server

再打开客户端窗口： src/redis-cli

redis> set foo bar 输出 OK

redis> get foo 输出 "bar"

## 2 生产环境部署

第一步：下载

wget http://download.redis.io/releases/redis-4.0.10.tar.gz

tar xzf redis-4.0.10.tar.gz

mv redis-4.0.10/ /usr/local/redis/

第二步：安装测试

cd /usr/local/redis/

make

make test //测试依赖

make install //将命令安装在 /usr/local/bin下

cd /usr/local/bin

ls -all //查看命令是否安装在本目录下

注意：make test不通过出现tcl问题解决办法如下

wget http://downloads.sourceforge.net/tcl/tcl8.6.1-src.tar.gz

sudo tar xzvf tcl8.6.1-src.tar.gz -C /usr/local/

cd /usr/local/tcl8.6.1/unix/

sudo ./configure

sudo make

sudo make install

第三步：移动配置文件

mkdir /etc/redis/

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

第四步：配置

bind 127.0.0.1 要绑定的ip

port 6379 要绑定的端口

daemonize 是否以守护进程开启

dbfilename 数据文件，默认为dump.rdb

dir 数据存储路径

database 数据库数，默认16个

slaveof 主从复制，类似双机备份

第五步：允许远程连接

**5.1 6379端口没有开放**

firewall-cmd --zone=public --add-port6379/tcp --permanent

--zone #作用域

--add-port=80/tcp  #添加端口，格式为：端口/通讯协议

--permanent  #永久生效，没有此参数重启后失效

5.2 关闭防火墙

6379开放后：

IMG_256

**2.修改redis.conf配置文件**

这个文件需要修改两个地方

* 打开redis.conf文件

　　按crrl+F查询，

* 找到bind 127.0.0.1，把这行前面加个#注释掉
* 再查找protected-mode yes 把yes修改为no，然后：wq！保存文件，
* 把服务关掉

src/redis-cli -h 127.0.0.1 -p 6379 shutdown

* 重启服务，后面加--protected-mode no

src/redis-server ----protected-mode no

## 3 配置文件样例

# 1k => 1000 bytes

# 1kb => 1024 bytes

# 1m => 1000000 bytes

# 1mb => 1024\*1024 bytes

# 1g => 1000000000 bytes

# 1gb => 1024\*1024\*1024 bytes

单位中不区分大小写1GB 1Gb 1gB是一样的

后台运行，yes是后台运行，no前台运行，将输出，输出到终端（默认）

daemonize yes

如果daemonize参数为yes的话就会产生pid文件，一下是pid文件的定义

pidfile /usr/local/redis-master/run/redis.pid

监听的端口

port 6379

绑定监听的IP地址

bind 127.0.0.1

如果在本地调用redis可以直接用sock文件

unixsocket /tmp/redis.sock //sock文件的位置

unixsocketperm 755 //sock文件的权限

如果一个链接在N秒内是空闲的，就将其关闭

timeout 0

如果对方down了或者中间网络断了发送ACK到客户端在指定的时间内没有收到对方的回应就断开TCP链接（时间单位秒记），此参数会受到内核参数的影响，推荐配置60。

tcp-keepalive 0

指定输出消息的级别

# debug (调试级别，详细信息，信息量大)

# verbose (详细信息，信息量较大)

# notice (通知，生产环境推荐)

# warning (错误信息警告信息)

loglevel notice

日志输出文件，默认在前端运行的时候此key的默认值是stdout输出到终端，如果用守护进程运行此key的stdout的时候将日志输入到/dev/null，如果想记录日志，就必须为其指定logfile位置

logfile /var/log/redis.log

将日志记录的哦syslog

syslog-enabled no

指定syslog的身份

syslog-ident redis

指定syslog的级别，必须是LOCAL0-LOCAL7之间

syslog-facility local0

设置数据库的数量

databases 16

设置数据库的数量。默认数据库DB 0，你可以选择一个不同的per-connection的使用SELECT<dbid>这儿的DBID是一个介于0和'databases'-1

databases 16

2.快照配置

将DB保存到磁盘的规则定义（快照）

格式：save <seconds> <changes>

例子：save 900 1 //在900秒（15分钟）内如果至少有1个键值发生变化 就保存

save 300 10 //在300秒（6分钟）内如果至少有10个键值发生变化 就保存

save 900 1 //每一条表示一个存盘点

save 300 10

save 60 10000

如果启用如上的快照（RDB），在一个存盘点之后，可能磁盘会坏掉或者权限问题，redis将依然能正常工作

stop-writes-on-bgsave-error yes

是否将字符串用LZF压缩到.rdb 数据库中，如果想节省CPU资源可以将其设置成no，但是字符串存储在磁盘上占用空间会很大，默认是yes

rdbcompression yes

rdb文件的校验，如果校验将避免文件格式坏掉，如果不校验将在每次操作文件时要付出校验过程的资源新能，将此参数设置为no，将跳过校验

rdbchecksum yes

转储数据的文件名

dbfilename dump.rdb

redis的工作目录，它会将转储文件存储到这个目录下，并生成一个附加文件

dir /usr/local/redis-master/db

3.主从参数

如果本地是salve服务器那么配置该项

# slaveof <masterip> <masterport>

slaveof 127.0.0.1 65532

master的验证密码

masterauth <master-password>

当从主机脱离主的链接时，如果此值为yes当客户端查询从时，回响应客户端，如果是第一次同步回返回一个日期数据或这空值，如果设置为no，则返回“SYNC with master in progress”到INFO and SLAVEOF

slave-serve-stale-data yes

从服务器只读（默认）

slave-read-only yes

从发送ping到主的时间间隔(单位：秒)

repl-ping-slave-period 10

批量传输I / O超时和主数据或ping响应超时 默认60s 必须大于repl-ping-slave-period值

repl-timeout 60

此选项如果是“yes”那么Redis的使用数量较少的TCP数据包和更少的带宽将数据发送到，在从主机上延迟40毫秒（linux kernel中的40毫秒）出现。如果是no将在slave中减少延迟，但是流量使用回相对多一些，如果用多个从主机，此处建议设置成yes

repl-disable-tcp-nodelay no

从主机的优先级，如果当主主机挂了的时候，将从从主机中选取一个作为其他从机的主，首先优先级的数字最低的将成为主，0是一个特殊的级别，0将永远不会成为主。默认值是100.

slave-priority 100