# 简介

## 特点

Redis 是完全开源免费的，遵守 BSD 协议，是**一个高性能的 key-value 数据库**。

Redis 与其他 key - value 缓存产品有以下三个特点：

1. Redis **支持数据的持久化**，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。
2. Redis 不仅仅支持简单的 key-value 类型的数据，同时还提供 list，set，zset，hash 等数据结构的存储。**多样的数据结构**
3. Redis 支持数据的备份，即 **master-slave 模式的数据备份**。

## Redis 优势

1. **性能极高** – Redis 能读的速度是 110000 次 / s, 写的速度是 81000 次 / s 。
2. **丰富的数据类型** – Redis 支持二进制案例的 Strings, Lists, Hashes, Sets 及 Ordered Sets 数据类型操作。
3. 原子 – Redis 的所有操作都是**原子性**的，意思就是要么成功执行要么失败完全不执行。单个操作是原子性的。**多个操作也支持事务**，即原子性，通过 MULTI 和 EXEC 指令包起来。
4. 丰富的特性 – Redis 还支持 publish/subscribe, 通知, key 过期等等特性。

## 和其他Key-value存储的区别

1. Redis 有着更为**复杂的数据结构**并且提供对他们的**原子性操作**，这是一个不同于其他数据库的进化路径。Redis 的数据类型都是基于基本数据结构的同时对程序员透明，无需进行额外的抽象。**事务**
2. **Redis 运行在内存中但是可以持久化到磁盘**，所以在对不同数据集进行高速读写时需要权衡内存，因为数据量不能大于硬件内存。
3. 在内存数据库方面的另一个优点是，相比在磁盘上相同的复杂的数据结构，在内存中操作起来非常简单，这样 Redis 可以做很多内部复杂性很强的事情。

# Redis安装

## 下载redis（目录结构）



## 注册/卸载服务

进入到redis安装目录

1. 注册服务

redis-server --service-install redis.windows.conf --loglevel verbose

1. 卸载服务

redis-server –service-uninstall

1. 开启服务

redis-server –service-start

1. 关闭服务

redis-server –service-stop

## 启动服务端和客户端

1. 启动服务端 redis-server path/redis.windows.conf
2. 启动客户端 reids-cli -h 127.0.0.1 -p 6379
3. 可以在同一个服务器上部署多个redis服务

#redis-server /usr/local/redis/redis6380.conf

#redis-server /usr/local/redis/redis6381.conf

## 测试

1. 在客户端中输入 set myKey abc get myKey 会得到返回值 abc

# Redis配置

## Config get获取配置命令

1. 进入客户端界面
2. 获取单个配置信息：config get 设置名（比如“loglevel”）
3. 获取所有的配置信息：config get \*

## Config set编辑配置命令

1. config set 配置项名字 配置项值

如：config set loglevel “notice”

## 配置说明

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

1. 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

1. 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

1. 绑定的主机地址

bind 127.0.0.1

1. 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

1. 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

1. 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

1. 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

1. 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。

(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

(11) 指定本地数据库文件名，默认值为 dump.rdb

dbfilename dump.rdb

(12) 指定本地数据库存放目录

dir ./

(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>

(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>

(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

(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

(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>

(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

(19) 指定更新日志文件名，默认为 appendonly.aof

appendfilename appendonly.aof

(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

(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

(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

(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

(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

(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

(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

(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

(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

(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

(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