**Redis配置参数详解**

<http://blog.51yip.com/nosql/1724.html>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GENERAL \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 是否作为守护进程运行

daemonize yes

// 如以后台进程运行，则需指定一个pid，默认为/var/run/redis.pid

pidfile /var/run/redis.pid

// Redis默认监听端口

port 6379

tcp-backlog 511

// 客户端闲置多少秒后，断开连接

timeout 0

tcp-keepalive 0

// 日志记录等级，有4个可选值，debug，verbose，notice，warning

loglevel notice

// 指定日志输出的文件名，可设为/dev/null屏蔽日志

logfile ""

// 可用数据库数，默认值为16，默认数据库为0

databases 16

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SNAPSHOTTING 快照 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 保存数据到disk的策略

// 900 秒有 1 条改变保存到disk

save 900 1

// 300 秒有 10 条改变保存到disk

save 300 10

// 60 秒有 10000 条改变保存到disk

save 60 10000

stop-writes-on-bgsave-error yes

// 当dump .rdb数据库的时候是否压缩数据对象

rdbcompression yes

rdbchecksum yes

// 本地数据库文件名，默认值为dump.rdb

dbfilename dump.rdb

// 本地数据库存放路径，默认值为 ./

dir ./

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* REPLICATION Redis的复制配置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 当本机为从服务时，设置主服务的IP及端口

// slaveof <masterip> <masterport>

// 当本机为从服务时，设置主服务的连接密码

// masterauth <master-password>

// 当从库同主机失去连接或者复制正在进行，从机库有两种运行方式

// 1) 如果slave-serve-stale-data设置为yes(默认设置)，从库会继续相应客户端的请求

// 2) 如果slave-serve-stale-data是指为no，出去INFO和SLAVOF命令之外的任何请求都会返回一个错误"SYNC with master in progress"

slave-serve-stale-data yes

slave-read-only yes

repl-diskless-sync no

repl-diskless-sync-delay 5

// 从库会按照一个时间间隔向主库发送PINGs.可以通过repl-ping-slave-period设置这个时间间隔，默认是10秒

repl-ping-slave-period 10

// repl-timeout 设置主库批量数据传输时间或者ping回复时间间隔，默认值是60秒

// 一定要确保repl-timeout大于repl-ping-slave-period

repl-timeout 60

// 采用无延迟同步 默认no

repl-disable-tcp-nodelay yes

slave-priority 100

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SECURITY 安全 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 设置客户端连接后进行任何其他指定前需要使用的密码。

// 警告：因为redis速度相当快，所以在一台比较好的服务器下，一个外部的用户可以在一秒钟进行150K次的密码尝试，

这意味着你需要指定非常非常强大的密码来防止暴力破解

// requirepass foobared

// 命令重命名.

// 在一个共享环境下可以重命名相对危险的命令。比如把CONFIG重名为一个不容易猜测的字符。

// 举例:

// rename-command CONFIG b840fc02d524045429941cc15f59e41cb7be6c52

// 如果想删除一个命令，直接把它重命名为一个空字符""即可，如下：

// rename-command CONFIG ""

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LIMITS 约束 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 最大可用内存 maxmemory <bytes> 536870912，即512M

maxmemory 536870912

// 当内存达到最大值的时候Redis会选择删除哪些数据？有五种方式可供选择

//

// volatile-lru -> 利用LRU算法移除设置过过期时间的key (LRU:最近使用 Least Recently Used )

// allkeys-lru -> 利用LRU算法移除任何key

// volatile-random -> 移除设置过过期时间的随机key

// allkeys->random -> remove a random key, any key

// volatile-ttl -> 移除即将过期的key(minor TTL)

// noeviction -> 不移除任何可以，只是返回一个写错误

maxmemory-policy allkeys-lru

// LRU 和 minimal TTL 算法都不是精准的算法，但是相对精确的算法(为了节省内存)，随意你可以选择样本大小进行检测。

// Redis默认的灰选择3个样本进行检测，你可以通过maxmemory-samples进行设置

maxmemory-samples 3

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* APPEND ONLY MODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 启用aof持久化方式

// 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。默认值为no

appendonly yes

// 更新日志文件名，默认值为appendonly.aof

appendfilename "appendonly.aof"

// 收到写命令立即写入磁盘，最慢，保证完全的持久化

appendfsync always

// 每秒写入一次

appendfsync everysec

// 完全依赖OS，性能最好，持久化没保证

appendfsync no

// 部署在同一机器的redis实例，把auto-aof-rewrite打开，因为cluster环境下内存占用基本一致

#关闭在aof rewrite的时候对新的写操作进行fsync

no-appendfsync-on-rewrite yes

// Automatic rewrite of the append only file.

// AOF 自动重写

// 当AOF文件增长到一定大小的时候Redis能够调用 BGREWRITEAOF 对日志文件进行重写

//

// 它是这样工作的：Redis会记住上次进行些日志后文件的大小(如果从开机以来还没进行过重写，那日子大小在开机的时候确定)

//

// 基础大小会同现在的大小进行比较。如果现在的大小比基础大小大制定的百分比，重写功能将启动

// 同时需要指定一个最小大小用于AOF重写，这个用于阻止即使文件很小但是增长幅度很大也去重写AOF文件的情况

// 设置 percentage 为0就关闭这个特性

auto-aof-rewrite-percentage 100

auto-aof-rewrite-min-size 64mb

aof-load-truncated yes

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LUA SCRIPTING \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

lua-time-limit 5000

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* REDIS CLUSTER 集群\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 打开redis集群

cluster-enabled yes

// cluster配置文件(启动自动生成)

cluster-config-file nodes-6379.conf

// 节点互连超时的阀值

cluster-node-timeout 15000

cluster-slave-validity-factor 10

cluster-migration-barrier 1

// 集群兼容部分失败

cluster-require-full-coverage yes

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SLOW LOG \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Redis Slow Log 记录超过特定执行时间的命令。执行时间不包括I/O计算比如连接客户端，返回结果等，只是命令执行时间

//

// 可以通过两个参数设置slow log：一个是告诉Redis执行超过多少时间被记录的参数slowlog-log-slower-than(微妙)，

// 另一个是slow log 的长度。当一个新命令被记录的时候最早的命令将被从队列中移除

// 下面的时间以微妙微单位，因此1000000代表一分钟。

// 注意制定一个负数将关闭慢日志，而设置为0将强制每个命令都会记录

slowlog-log-slower-than 10000

// 对日志长度没有限制，只是要注意它会消耗内存

// 可以通过 SLOWLOG RESET 回收被慢日志消耗的内存

slowlog-max-len 128

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* LATENCY MONITOR \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

latency-monitor-threshold 0

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* EVENT NOTIFICATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

notify-keyspace-events ""

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADVANCED CONFIG \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 当hash中包含超过指定元素个数并且最大的元素没有超过临界时，

// hash将以一种特殊的编码方式（大大减少内存使用）来存储，这里可以设置这两个临界值

// Redis Hash对应Value内部实际就是一个HashMap，实际这里会有2种不同实现，

// 这个Hash的成员比较少时Redis为了节省内存会采用类似一维数组的方式来紧凑存储，

而不会采用真正的HashMap结构，对应的value redisObject的encoding为zipmap,

// 当成员数量增大时会自动转成真正的HashMap,此时encoding为ht。

hash-max-ziplist-entries 512

hash-max-ziplist-value 64

// list数据类型多少节点以下会采用去指针的紧凑存储格式。

// list数据类型节点值大小小于多少字节会采用紧凑存储格式。

list-max-ziplist-entries 512

list-max-ziplist-value 64

// set数据类型内部数据如果全部是数值型，且包含多少节点以下会采用紧凑格式存储。

set-max-intset-entries 512

// zsort数据类型多少节点以下会采用去指针的紧凑存储格式。

// zsort数据类型节点值大小小于多少字节会采用紧凑存储格式。

zset-max-ziplist-entries 128

zset-max-ziplist-value 64

hll-sparse-max-bytes 3000

// Redis将在每100毫秒时使用1毫秒的CPU时间来对redis的hash表进行重新hash，可以降低内存的使用

//

// 当你的使用场景中，有非常严格的实时性需要，不能够接受Redis时不时的对请求有2毫秒的延迟的话，把这项配置为no。

//

// 如果没有这么严格的实时性要求，可以设置为yes，以便能够尽可能快的释放内存

activerehashing yes

client-output-buffer-limit normal 0 0 0

client-output-buffer-limit slave 256mb 64mb 60

client-output-buffer-limit pubsub 32mb 8mb 60

hz 10

aof-rewrite-incremental-fsync yes

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* VM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 是否使用虚拟内存，默认值为no

vm-enabled yes

// 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

// 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

// 虚拟内存文件以块存储，每块32bytes

vm-page-size 32

// 虚拟内在文件的最大数

vm-pages 134217728

// 可以设置访问swap文件的线程数,设置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,但是对数据完整性有很好的保证.

vm-max-threads 4

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INCLUDES \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 包含通用配置

include /etc/redis/redis-common.conf

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* GENERAL \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 如以后台进程运行，则需指定一个pid，默认为/var/run/redis.pid

pidfile /var/run/redis\_6379.pid

// Redis默认监听端口

port 6379

// 指定日志输出的文件名，可设为/dev/null屏蔽日志

logfile /var/log/redis\_6379.log

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SNAPSHOTTING 快照 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 本地数据库文件名，默认值为dump.rdb

dbfilename dump6379.rdb

// 本地数据库存放路径，默认值为 ./

dir /var/redis/6379

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* REPLICATION Redis的复制配置 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 当本机为从服务时，设置主服务的IP及端口

// slaveof <masterip> <masterport>

// 当本机为从服务时，设置主服务的连接密码

// masterauth <master-password>

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* APPEND ONLY MODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// 更新日志文件名，默认值为appendonly.aof

appendfilename "appendonly6379.aof"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* REDIS CLUSTER 集群 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// cluster配置文件(启动自动生成)

cluster-config-file nodes-6379.conf