

## redis 参数优化

vim /etc/sysctl.conf

vm.overcommit\_memory = 1

#最好不要设置为0，Linux的OOM机制在内存不足的情况下，会自动选择性Kill进程点数过高的进程，0会中招。

sysctl vm.overcommit\_memory=1

0：表示内核将检查是否有足够的可用内存供应用进程使用；如果有足够的可用内存，内存申请允许；否则，内存申请失败，并把错误返回给应用进程。

1：表示内核允许分配所有的物理内存，而不管当前的内存状态如何。

2：表示内核允许分配超过所有物理内存和交换空间总和的内存

vim /etc/redis/redis.conf

#####基础#####

daemonize yes //是否以守护进程方式启动

pidfile /var/run/redis.pid //如果要玩单机多开，需要设定不同path的pid，除非硬件特别紧张，一般不会

port 6379 //端口

tcp-backlog 600 //如果服务器caps很高，需要把这个参数改大些。

timeout 0 //客户端连接的超时时间,单位为秒,超时后会关闭连接，0永不超时

tcp-keepalive 0 //检测挂掉的连接，单位s，0禁止

loglevel notice //日志记录等级，4个可选值 debug(调试) > verbose(精简) > notice(适量) > warning(警告)

logfile "./redis.log" //日志文件路径

databases 6 //控制数据库的总数默认客户端进0号

#####RDB

快

照

#####

save 900 1 // 900秒（15分钟）之后，且至少1次变更频率不要太高，  
RDB快照代价比较高

save 300 10 //300秒（5分钟）之后，且至少10次变更频率不要太高，  
RDB快照代价比较高

save 60 1000

rdbcompression yes //快照时压缩

dbfilename dump.rdb //快照名

dir ./ //快照文件路径

stop-writes-on-bgsave-error yes //save失败(快照)是否停止写操作，如  
果有监控机制可以no

rdbchecksum yes //生成和加载的时候是否开启CRC64检查，开启后性  
能消耗大概10%，但更加安全

##### 复 制 ( 从 库 配  
置)#####

slaveof ip地址 6379 //一旦配置(从配置，主不需要)，则开启主从复制

masterauth howbuy //配置主的密码

slave-serve-stale-data no //从丢失主，或者同步过程中，是否可继续响  
应客户端

repl-disable-tcp-nodelay no //yes 合并tcp包节省带宽，但产生40ms左  
右的延时。no 立马发送数据，无延迟。从太多，可以考虑yes

slave-priority 100 主挂了，从变主的概率，越小概率越高，但0表示永  
不做主机

#####安全#####

requirepass howbuy //设置密码

#####

限

制

#####

maxclients 128 //限制客户端的连接数

maxmemory 2gb //最大内存最好不要超过空闲内存的3/5，超过32GB会自动进入64位世界，指针长度\*2，20%的空间会被指针消耗，性能略有影响

maxmemory-policy allkeys-lru //4种策略 volatile-lru、allkeys-lru、volatile-random、allkeys-random、noeviction根据LRU算法移除内存中所有的key

maxmemory-samples 5 //五个key然后取最旧的那个，LRU和最小TTL算法的优化

lua-time-limit 5000 //一个Lua脚本最长的执行时间为5000毫秒,0或负数表示无限执行时间

#####

APPEND

ONLY

MODE

(AOF)#####

appendonly yes //是否开启AOF

appendfilename "appendonly.aof" //AOF文件名

appendfsync everysec //每秒写入性能 no(操作系统决定)  
>everysec(每秒写一次) > always(每次立马写入)

no-appendfsync-on-rewrite yes //写AOF的时候放弃同步主进程的变化，可能会丢日志，但是在高并发的时候不会出现一卡一卡的现象

auto-aof-rewrite-percentage 100 //AOF文件体积扩大100%的时候重写该日志

auto-aof-rewrite-min-size 64mb //除了百分比，再加体积限制

aof-load-truncated yes //redis在启动时可以加载被截断的AOF文件

aof-rewrite-incremental-fsync yes //当修改AOF文件时，该设置为yes，则每生成32MB的数据，就进行同步

##### 慢 查 日 志

#####

slowlog-log-slower-than 1000000 //记录超过1秒的操作

slowlog-max-len 50 //记录50个

##### 虚拟内存 #####

#vm-enabled no //不用虚拟内存，太穷的话就没必要用redis

#####高级配置#####

hash-max-ziplist-entries 512 //配置最大元素数，当超过该配置数据时，redis采用特殊hash算法

hash-max-ziplist-value 64 //配置最大元素值，当超过配置值时，采用特殊hash算法

list-max-ziplist-entries 512

list-max-ziplist-value 64

set-max-intset-entries 512

set-max-intset-value 64

zset-max-ziplist-entries 128

zset-max-ziplist-value 64

hll-sparse-max-bytes 5000 //配置超重对数基数，CPU比较牛的情况下可以上10000

activeremhashing yes //哈希刷新，如果你不太在意延迟而希望尽快释放内存的话就设置

client-output-buffer-limit normal 0 0 0 //对客户端输出缓冲进行限制可以强迫那些就不从服务器读取数据的客户端断开连接。对于normalclient，第一个0表示取消hardlimit，第二个0和第三个0表示取消

soft limit , normalclient默认取消限制 , 因为如果没有寻问 , 他们是不会接收数据的

client-output-buffer-limit slave 256mb 64mb 60 //对于slaveclient和MONITER client , 如果client-output-buffer一旦超过256mb , 又或者超过64mb持续60秒 , 那么服务器就会立即断开客户端连接。

client-output-buffer-limit pubsub 32mb 8mb 60 //对于pubsub client , 如果client-output-buffer一旦超过32mb , 又或者超过8mb持续60秒 , 那么服务器就会立即断开客户端连接。

hz 20 #redis内部调度 ( 进行关闭timeout的客户端 , 删除过期key等等 ) 频率

```
##### LATENCY
MONITOR#####
```

latency-monitor-threshold 0 //用LATENCY打印redis实例在跑命令时的耗时图表 , 监视频率 , 0为不监视

## #重要参数

daemonize yes #守护进程模式

save 60 1000 #当时间间隔超过60秒 , 或存储超过1000条记录时 , 进行持久化。

maxmemory 256mb #分配256MB内存

#注意 : 如果是主库挂了 , 先把从库的dump文件拷贝到主库/etc/redis目录下在 , 再启动主库。

#scp /etc/redis/dump.rdb root@ip地址:/etc/redis/dump.rdb