**Redis 配置文件详解**

redis 是一款开源的、高性能的键-值存储（key-value store），和 memcached 类似，redis 常被称作

是一款 key-value 内存存储系统或者内存数据库，同时由于它支持丰富的数据结构，又被称为一种数据结

构服务器（data structure server）。

编译完 redis，它的配置文件在源码目录下 redis.conf ，将其拷贝到工作目录下即可使用，下面具体解释

redis.conf 中的各个参数：

**1 daemonize no**

默认情况下，redis 不是在后台运行的，如果需要在后台运行，把该项的值更改为 yes。

**2 pidfile /var/run/redis.pid**

当 Redis 在后台运行的时候，Redis 默认会把 pid 文件放在/var/run/redis.pid，你可以

配置到其他地址。当运行多个 redis 服务时，需要指定不同的 pid 文件和端口

**3 port**

监听端口，默认为 6379

**4 #bind 127.0.0.1**

指定 Redis 只接收来自于该 IP 地址的请求，如果不进行设置，那么将处理所有请求，在生

产环境中为了安全最好设置该项。默认注释掉，不开启

**5 timeout 0**

设置客户端连接时的超时时间，单位为秒。当客户端在这段时间内没有发出任何指令，那么

关闭该连接

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

**6 tcp-keepalive 0**

指定 TCP 连接是否为长连接,"侦探"信号有 server 端维护。默认为 0.表示禁用

**7 loglevel notice**

log 等级分为 4 级，debug,verbose, notice, 和 warning。生产环境下一般开启 notice

**8 logfile stdout**

配置 log 文件地址，默认使用标准输出，即打印在命令行终端的窗口上，修改为日志文件

目录

**9 databases 16**

设置数据库的个数，可以使用 SELECT 命令来切换数据库。默认使用的数据库是 0 号库。

默认 16 个库

**10**

**save 900 1**

**save 300 10**

**save 60 10000**

保存数据快照的频率，即将数据持久化到 dump.rdb 文件中的频度。用来描述"在多少秒期

间至少多少个变更操作"触发 snapshot 数据保存动作

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

默认设置，意思是：

if(在 60 秒之内有 10000 个 keys 发生变化时){

进行镜像备份

}else if(在 300 秒之内有 10 个 keys 发生了变化){

进行镜像备份

}else if(在 900 秒之内有 1 个 keys 发生了变化){

进行镜像备份

}

**11 stop-writes-on-bgsave-error yes**

当持久化出现错误时，是否依然继续进行工作，是否终止所有的客户端 write 请求。默认设

置"yes"表示终止，一旦 snapshot 数据保存故障，那么此 server 为只读服务。如果为"no"，

那么此次 snapshot 将失败，但下一次 snapshot 不会受到影响，不过如果出现故障,数据只

能恢复到"最近一个成功点"

**12 rdbcompression yes**

在进行数据镜像备份时，是否启用 rdb 文件压缩手段，默认为 yes。压缩可能需要额外的

cpu 开支，不过这能够有效的减小 rdb 文件的大，有利于存储/备份/传输/数据恢复

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

**13 rdbchecksum yes**

读取和写入时候，会损失 10%性能

是否进行校验和，是否对 rdb 文件使用 CRC64 校验和,默认为"yes"，那么每个 rdb 文件内

容的末尾都会追加 CRC 校验和，利于第三方校验工具检测文件完整性

**14 dbfilename dump.rdb**

镜像备份文件的文件名，默认为 dump.rdb

**15 dir ./**

数据库镜像备份的文件 rdb/AOF 文件放置的路径。这里的路径跟文件名要分开配置是因为

Redis 在进行备份时，先会将当前数据库的状态写入到一个临时文件中，等备份完成时，

再把该临时文件替换为上面所指定的文件，而这里的临时文件和上面所配置的备份文件都会

放在这个指定的路径当中

**16 # slaveof <masterip> <masterport>**

设置该数据库为其他数据库的从数据库，并为其指定 master 信息。

**17 masterauth**

当主数据库连接需要密码验证时，在这里指定

**18 slave-serve-stale-data yes**

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

当主master服务器挂机或主从复制在进行时，是否依然可以允许客户访问可能过期的数据。

在"yes"情况下,slave 继续向客户端提供只读服务,有可能此时的数据已经过期；在"no"情况

下，任何向此 server 发送的数据请求服务(包括客户端和此 server 的 slave)都将被告知

"error"

**19 slave-read-only yes**

slave 是否为"只读"，强烈建议为"yes"

**20 # repl-ping-slave-period 10**

slave 向指定的 master 发送 ping 消息的时间间隔(秒)，默认为 10

**21 # repl-timeout 60**

slave 与 master 通讯中,最大空闲时间,默认 60 秒.超时将导致连接关闭

**22 repl-disable-tcp-nodelay no**

slave 与 master 的连接,是否禁用 TCP nodelay 选项。"yes"表示禁用,那么 socket 通讯中

数据将会以 packet 方式发送(packet 大小受到 socket buffer 限制)。

可以提高 socket通讯的效率(tcp交互次数),但是小数据将会被buffer,不会被立即发送,对于

接受者可能存在延迟。"no"表示开启 tcp nodelay 选项,任何数据都会被立即发送,及时性较

好,但是效率较低，建议设为 no

**23 slave-priority 100**

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

适用 Sentinel 模块(unstable,M-S 集群管理和监控),需要额外的配置文件支持。slave 的权

重值,默认100.当master失效后,Sentinel将会从slave列表中找到权重值最低(>0)的slave,

并提升为 master。如果权重值为 0,表示此 slave 为"观察者",不参与 master 选举

**24 # requirepass foobared**

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。

**25 # rename-command CONFIG 3ed984507a5dcd722aeade310065ce5d**

(方

式:MD5('CONFIG^!'))

重命名指令,对于一些与"server"控制有关的指令,可能不希望远程客户端(非管理员用户)链

接随意使用,那么就可以把这些指令重命名为"难以阅读"的其他字符串

**26 # maxclients 10000**

限制同时连接的客户数量。当连接数超过这个值时，redis 将不再接收其他连接请求，客户

端尝试连接时将收到 error 信息。默认为 10000，要考虑系统文件描述符限制，不宜过大，

浪费文件描述符，具体多少根据具体情况而定

**27 # maxmemory <bytes>**

redis-cache 所能使用的最大内存(bytes),默认为 0,表示"无限制",最终由 OS 物理内存大小

决定(如果物理内存不足,有可能会使用 swap)。此值尽量不要超过机器的物理内存尺寸,从性

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

能和实施的角度考虑,可以为物理内存 3/4。此配置需要和"maxmemory-policy"配合使用,

当 redis 中内存数据达到 maxmemory 时,触发"清除策略"。在"内存不足"时,任何 write 操

作(比如 set,lpush 等)都会触发"清除策略"的执行。在实际环境中,建议 redis 的所有物理机

器的硬件配置保持一致(内存一致),同时确保 master/slave 中"maxmemory""policy"配置

一致。

当内存满了的时候，如果还接收到 set 命令，redis 将先尝试剔除设置过 expire 信息的

key，而不管该 key 的过期时间还没有到达。在删除时，

将按照过期时间进行删除，最早将要被过期的 key 将最先被删除。如果带有 expire 信息

的 key 都删光了，内存还不够用，那么将返回错误。这样，redis 将不再接收写请求，只

接收 get 请求。maxmemory 的设置比较适合于把 redis 当作于类似 memcached 的缓

存来使用。

**28 # maxmemory-policy volatile-lru**

内存不足"时,数据清除策略,默认为"volatile-lru"。

volatile-lru ->对"过期集合"中的数据采取 LRU(近期最少使用)算法.如果对 key 使用

"expire"指令指定了过期时间,那么此 key 将会被添加到"过期集合"中。将已经过期/LRU 的

数据优先移除.如果"过期集合"中全部移除仍不能满足内存需求,将 OOM.

allkeys-lru ->对所有的数据,采用 LRU 算法

volatile-random ->对"过期集合"中的数据采取"随即选取"算法,并移除选中的 K-V,直到"

内存足够"为止. 如果如果"过期集合"中全部移除全部移除仍不能满足,将 OOM

allkeys-random ->对所有的数据,采取"随机选取"算法,并移除选中的 K-V,直到"内存足够"

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

为止

volatile-ttl ->对"过期集合"中的数据采取 TTL 算法(最小存活时间),移除即将过期的数据.

noeviction ->不做任何干扰操作,直接返回 OOM 异常

另外，如果数据的过期不会对"应用系统"带来异常,且系统中 write 操作比较密集,建议采取

"allkeys-lru"

**29 # maxmemory-samples 3**

默认值 3，上面 LRU 和最小 TTL 策略并非严谨的策略，而是大约估算的方式，因此可以选

择取样值以便检查

**29 appendonly no**

默认情况下，redis 会在后台异步的把数据库镜像备份到磁盘，但是该备份是非常耗时的，

而且备份也不能很频繁。所以 redis 提供了另外一种更加高效的数据库备份及灾难恢复方

式。开启 append only 模式之后，redis 会把所接收到的每一次写操作请求都追加到

appendonly.aof 文件中，当 redis 重新启动时，会从该文件恢复出之前的状态。但是这样

会造成 appendonly.aof 文件过大，所以 redis 还支持了 BGREWRITEAOF 指令，对

appendonly.aof 进行重新整理。如果不经常进行数据迁移操作，推荐生产环境下的做法为

关闭镜像，开启 appendonly.aof，同时可以选择在访问较少的时间每天对 appendonly.aof

进行重写一次。

另外，对 master 机器,主要负责写，建议使用 AOF,对于 slave,主要负责读，挑选出 1-2 台

开启 AOF，其余的建议关闭

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

**30 # appendfilename appendonly.aof**

aof 文件名字，默认为 appendonly.aof

**31 appendfsync aof 同步频率**

**# appendfsync always**

**appendfsync everysec**

**# appendfsync no**

设置对 appendonly.aof 文件进行同步的频率。always 表示每次有写操作都进行同步，

everysec 表示对写操作进行累积，每秒同步一次。no 不主动 fsync，由 OS 自己来完成。

这个需要根据实际业务场景进行配置

**32 no-appendfsync-on-rewrite no**

在 aof rewrite 期间,是否对 aof 新记录的 append 暂缓使用文件同步策略,主要考虑磁盘 IO

开支和请求阻塞时间。默认为 no,表示"不暂缓",新的 aof 记录仍然会被立即同步

**33 auto-aof-rewrite-percentage 100**

当 Aof log 增长超过指定比例时，重写 log file， 设置为 0 表示不自动重写 Aof 日志，重

写是为了使 aof 体积保持最小，而确保保存最完整的数据。

**34 auto-aof-rewrite-min-size 64mb**

触发 aof rewrite 的最小文件尺寸

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

**35 lua-time-limit 5000**

lua 脚本运行的最大时间

**36 slowlog-log-slower-than 10000**

"慢操作日志"记录,单位:微秒(百万分之一秒,1000 \* 1000),如果操作时间超过此值,将会把

command 信息"记录"起来.(内存,非文件)。其中"操作时间"不包括网络 IO 开支,只包括请求

达到 server 后进行"内存实施"的时间."0"表示记录全部操作

**37 slowlog-max-len 128**

"慢操作日志"保留的最大条数,"记录"将会被队列化,如果超过了此长度,旧记录将会被移除。

可以通过"SLOWLOG <subcommand> args"查看慢记录的信息(SLOWLOG get

10,SLOWLOG reset)

**38**

**hash-max-ziplist-entries 512**

hash 类型的数据结构在编码上可以使用 ziplist 和 hashtable。ziplist 的特点就是文件存储

(以及内存存储)所需的空间较小,在内容较小时,性能和 hashtable 几乎一样.因此 redis 对

hash 类型默认采取 ziplist。如果 hash 中条目的条目个数或者 value 长度达到阀值,将会被

重构为 hashtable。

这个参数指的是 ziplist 中允许存储的最大条目个数，，默认为 512，建议为 128

**hash-max-ziplist-value 64**

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

ziplist 中允许条目 value 值最大字节数，默认为 64，建议为 1024

**39**

**list-max-ziplist-entries 512**

**list-max-ziplist-value 64**

对于 list 类型,将会采取 ziplist,linkedlist 两种编码类型。解释同上。

**40 set-max-intset-entries 512**

intset 中允许保存的最大条目个数,如果达到阀值,intset 将会被重构为 hashtable

**41**

**zset-max-ziplist-entries 128**

**zset-max-ziplist-value 64**

zset 为有序集合,有 2 中编码类型:ziplist,skiplist。因为"排序"将会消耗额外的性能,当 zset

中数据较多时,将会被重构为 skiplist。

**42 activerehashing yes**

是否开启顶层数据结构的 rehash 功能,如果内存允许,请开启。rehash 能够很大程度上提高

K-V 存取的效率

**43**

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

**client-output-buffer-limit normal 0 0 0**

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

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

客户端 buffer 控制。在客户端与 server 进行的交互中,每个连接都会与一个 buffer 关联,此

buffer 用来队列化等待被 client 接受的响应信息。如果 client 不能及时的消费响应信息,那

么 buffer 将会被不断积压而给 server 带来内存压力.如果 buffer 中积压的数据达到阀值,将

会导致连接被关闭,buffer 被移除。

buffer 控制类型包括:normal -> 普通连接；slave ->与 slave 之间的连接；

pubsub ->pub/sub 类型连接，此类型的连接，往往会产生此种问题;因为 pub 端会密集的

发布消息,但是 sub 端可能消费不足.

指令格式:client-output-buffer-limit <class> <hard> <soft> <seconds>",其中 hard

表示 buffer 最大值,一旦达到阀值将立即关闭连接;

soft 表示"容忍值",它和 seconds 配合,如果 buffer 值超过 soft 且持续时间达到了 seconds,

也将立即关闭连接,如果超过了 soft 但是在 seconds 之后，buffer 数据小于了 soft,连接将

会被保留.

其中 hard 和 soft 都设置为 0,则表示禁用 buffer 控制.通常 hard 值大于 soft.

**44 hz 10**

Redis server 执行后台任务的频率,默认为 10,此值越大表示 redis 对"间歇性 task"的执行次

数越频繁(次数/秒)。"间歇性 task"包括"过期集合"检测、关闭"空闲超时"的连接等,此值必

你只有非常努力，才能看起来毫不费力！

qq: 2298535141



**Redis 配置文件详解**

须大于 0 且小于 500。此值过小就意味着更多的 cpu 周期消耗,后台 task 被轮询的次数更频

繁。此值过大意味着"内存敏感"性较差。建议采用默认值。

**45 include 载入文件**

**# include /path/to/local.conf**

**# include /path/to/other.conf**

额外载入配置文件。

你只有非常努力，才能看起来毫不费力！

qq: 2298535141

