**Redis windows 测试redis持久化功能**

还需要在redis根目录增加一个redis的配置文件redis.conf，文件具体内容有：

1. *# Redis configuration file example*
2.
3. *# By default Redis does not run as a daemon. Use 'yes' if you need it.*
4. *# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.*
5. daemonize no
6.
7. *# When run as a daemon, Redis write a pid file in /var/run/redis.pid by default.*
8. *# You can specify a custom pid file location here.*
9. pidfile /var/run/redis.pid
10.
11. *# Accept connections on the specified port, default is 6379*
12. port 6379
13.
14. *# If you want you can bind a single interface, if the bind option is not*
15. *# specified all the interfaces will listen for connections.*
16. *#*
17. *# bind 127.0.0.1*
18.
19. *# Close the connection after a client is idle for N seconds (0 to disable)*
20. timeout 300
21.
22. *# Set server verbosity to 'debug'*
23. *# it can be one of:*
24. *# debug (a lot of information, useful for development/testing)*
25. *# notice (moderately verbose, what you want in production probably)*
26. *# warning (only very important / critical messages are logged)*
27. loglevel debug
28.
29. *# Specify the log file name. Also 'stdout' can be used to force*
30. *# the demon to log on the standard output. Note that if you use standard*
31. *# output for logging but daemonize, logs will be sent to /dev/null*
32. logfile stdout
33.
34. *# Set the number of databases. The default database is DB 0, you can select*
35. *# a different one on a per-connection basis using SELECT <dbid> where*
36. *# dbid is a number between 0 and 'databases'-1*
37. databases 16
38.
39. *################################ SNAPSHOTTING #################################*
40. *#*
41. *# Save the DB on disk:*
42. *#*
43. *# save <seconds> <changes>*
44. *#*
45. *# Will save the DB if both the given number of seconds and the given*
46. *# number of write operations against the DB occurred.*
47. *#*
48. *# In the example below the behaviour will be to save:*
49. *# after 900 sec (15 min) if at least 1 key changed*
50. *# after 300 sec (5 min) if at least 10 keys changed*
51. *# after 60 sec if at least 10000 keys changed*
52. save 900 1
53. save 300 10
54. save 60 10000
55.
56. *# Compress string objects using LZF when dump .rdb databases?*
57. *# For default that's set to 'yes' as it's almost always a win.*
58. *# If you want to save some CPU in the saving child set it to 'no' but*
59. *# the dataset will likely be bigger if you have compressible values or keys.*
60. rdbcompression yes
61.
62. *# The filename where to dump the DB*
63. dbfilename dump.rdb
64.
65. *# For default save/load DB in/from the working directory*
66. *# Note that you must specify a directory not a file name.*
67. dir ./
68.
69. *################################# REPLICATION #################################*
70.
71. *# Master-Slave replication. Use slaveof to make a Redis instance a copy of*
72. *# another Redis server. Note that the configuration is local to the slave*
73. *# so for example it is possible to configure the slave to save the DB with a*
74. *# different interval, or to listen to another port, and so on.*
75. *#*
76. *# slaveof <masterip> <masterport>*
77.
78. *# If the master is password protected (using the "requirepass" configuration*
79. *# directive below) it is possible to tell the slave to authenticate before*
80. *# starting the replication synchronization process, otherwise the master will*
81. *# refuse the slave request.*
82. *#*
83. *# masterauth <master-password>*
84.
85. *################################## SECURITY ###################################*
86.
87. *# Require clients to issue AUTH <PASSWORD> before processing any other*
88. *# commands. This might be useful in environments in which you do not trust*
89. *# others with access to the host running redis-server.*
90. *#*
91. *# This should stay commented out for backward compatibility and because most*
92. *# people do not need auth (e.g. they run their own servers).*
93. *#*
94. *# requirepass foobared*
95.
96. *################################### LIMITS ####################################*
97.
98. *# Set the max number of connected clients at the same time. By default there*
99. *# is no limit, and it's up to the number of file descriptors the Redis process*
100. *# is able to open. The special value '0' means no limts.*
101. *# Once the limit is reached Redis will close all the new connections sending*
102. *# an error 'max number of clients reached'.*
103. *#*
104. *# maxclients 128*
105.
106. *# Don't use more memory than the specified amount of bytes.*
107. *# When the memory limit is reached Redis will try to remove keys with an*
108. *# EXPIRE set. It will try to start freeing keys that are going to expire*
109. *# in little time and preserve keys with a longer time to live.*
110. *# Redis will also try to remove objects from free lists if possible.*
111. *#*
112. *# If all this fails, Redis will start to reply with errors to commands*
113. *# that will use more memory, like SET, LPUSH, and so on, and will continue*
114. *# to reply to most read-only commands like GET.*
115. *#*
116. *# WARNING: maxmemory can be a good idea mainly if you want to use Redis as a*
117. *# 'state' server or cache, not as a real DB. When Redis is used as a real*
118. *# database the memory usage will grow over the weeks, it will be obvious if*
119. *# it is going to use too much memory in the long run, and you'll have the time*
120. *# to upgrade. With maxmemory after the limit is reached you'll start to get*
121. *# errors for write operations, and this may even lead to DB inconsistency.*
122. *#*
123. *# maxmemory <bytes>*
124.
125. *############################## APPEND ONLY MODE ###############################*
126.
127. *# By default Redis asynchronously dumps the dataset on disk. If you can live*
128. *# with the idea that the latest records will be lost if something like a crash*
129. *# happens this is the preferred way to run Redis. If instead you care a lot*
130. *# about your data and don't want to that a single record can get lost you should*
131. *# enable the append only mode: when this mode is enabled Redis will append*
132. *# every write operation received in the file appendonly.log. This file will*
133. *# be read on startup in order to rebuild the full dataset in memory.*
134. *#*
135. *# Note that you can have both the async dumps and the append only file if you*
136. *# like (you have to comment the "save" statements above to disable the dumps).*
137. *# Still if append only mode is enabled Redis will load the data from the*
138. *# log file at startup ignoring the dump.rdb file.*
139. *#*
140. *# The name of the append only file is "appendonly.log"*
141. *#*
142. *# IMPORTANT: Check the BGREWRITEAOF to check how to rewrite the append*
143. *# log file in background when it gets too big.*
144.
145. appendonly yes
146.
147. *# The fsync() call tells the Operating System to actually write data on disk*
148. *# instead to wait for more data in the output buffer. Some OS will really flush*
149. *# data on disk, some other OS will just try to do it ASAP.*
150. *#*
151. *# Redis supports three different modes:*
152. *#*
153. *# no: don't fsync, just let the OS flush the data when it wants. Faster.*
154. *# always: fsync after every write to the append only log . Slow, Safest.*
155. *# everysec: fsync only if one second passed since the last fsync. Compromise.*
156. *#*
157. *# The default is "always" that's the safer of the options. It's up to you to*
158. *# understand if you can relax this to "everysec" that will fsync every second*
159. *# or to "no" that will let the operating system flush the output buffer when*
160. *# it want, for better performances (but if you can live with the idea of*
161. *# some data loss consider the default persistence mode that's snapshotting).*
162.
163. appendfsync always
164. *# appendfsync everysec*
165. *# appendfsync no*
166.
167. *############################### ADVANCED CONFIG ###############################*
168.
169. *# Glue small output buffers together in order to send small replies in a*
170. *# single TCP packet. Uses a bit more CPU but most of the times it is a win*
171. *# in terms of number of queries per second. Use 'yes' if unsure.*
172. glueoutputbuf yes
173.
174. *# Use object sharing. Can save a lot of memory if you have many common*
175. *# string in your dataset, but performs lookups against the shared objects*
176. *# pool so it uses more CPU and can be a bit slower. Usually it's a good*
177. *# idea.*
178. *#*
179. *# When object sharing is enabled (shareobjects yes) you can use*
180. *# shareobjectspoolsize to control the size of the pool used in order to try*
181. *# object sharing. A bigger pool size will lead to better sharing capabilities.*
182. *# In general you want this value to be at least the double of the number of*
183. *# very common strings you have in your dataset.*
184. *#*
185. *# WARNING: object sharing is experimental, don't enable this feature*
186. *# in production before of Redis 1.0-stable. Still please try this feature in*
187. *# your development environment so that we can test it better.*
188. *# shareobjects no*
189. *# shareobjectspoolsize 1024*

启动redis

打开运行窗口

F:\>cd redis-2.0.2

F:\redis-2.0.2>redis-server.exe redis.conf
[2944] 15 Jun 22:44:29 \* Server started, Redis version 2.0.2
[2944] 15 Jun 22:44:29 \* DB loaded from append only file: 0 seconds
[2944] 15 Jun 22:44:29 \* The server is now ready to accept connections on port
379
[2944] 15 Jun 22:44:30 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[2944] 15 Jun 22:44:30 - 0 clients connected (0 slaves), 450888 bytes in use

在打开一个窗口运行客户端

F:\redis-2.0.2>redis-cli.exe

redis>

设置值：

redis> set ajun ajun

Reconnecting... OK
OK

取值：

redis> get ajun
"ajun"

停止redis服务

redis> shutdown

如果需要redis持久化数据 需要配置redis日志开启

在每次更新操作后进行日志记录，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。默认值为no

此时要在redis.conf中修改或者添加

appendonly yes

更新日志文件名，默认值为appendonly.aof

#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，always表示每次更新操作后手动调用fsync()将数据写到磁盘，everysec表示每秒同步一次（默认值）。
# appendfsync always
**appendfsync everysec**
# appendfsync no

关闭redis 服务在重启

F:\redis-2.0.2>redis-server.exe redis.conf
[2944] 15 Jun 22:44:29 \* Server started, Redis version 2.0.2
[2944] 15 Jun 22:44:29 \* DB loaded from append only file: 0 seconds
[2944] 15 Jun 22:44:29 \* The server is now ready to accept connections on port
379
[2944] 15 Jun 22:44:30 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[2944] 15 Jun 22:44:30 - 0 clients connected (0 slaves), 450888 bytes in use
此时redis根目录会有产生一个appendonly.aof的文件来记录日志
在客户端重新连接

F:\redis-2.0.2>redis-cli.exe

redis>set ajun wahaha

然后在shutdown redis服务

查看appendonly.aof为1k

再启动redis服务

F:\redis-2.0.2>redis-server.exe redis.conf
[2944] 15 Jun 22:44:29 \* Server started, Redis version 2.0.2
[2944] 15 Jun 22:44:29 \* DB loaded from append only file: 0 seconds
[2944] 15 Jun 22:44:29 \* The server is now ready to accept connections on port
379
[2944] 15 Jun 22:44:30 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[2944] 15 Jun 22:44:30 - 0 clients connected (0 slaves), 450888 bytes in use

再启动客户端

F:\redis-2.0.2>redis-cli.exe

redis>get ajun
"wahaha"

值还在，说明被持久化了

linux上的操作也是类似的

reids.conf参数配置参考，具体看官方配置文件参数注解

**1. redis.conf配置参数：**

#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 redis.pid**
#绑定主机IP，默认值为127.0.0.1
#bind 127.0.0.1
#Redis默认监听端口
**port 6379**
#客户端闲置多少秒后，断开连接，默认为300（秒）
**timeout 300**
#日志记录等级，有4个可选值，debug，verbose（默认值），notice，warning
**loglevel verbose**
#指定日志输出的文件名，默认值为stdout，也可设为/dev/null屏蔽日志
**logfile stdout**
#可用数据库数，默认值为16，默认数据库为0
**databases 16**
#保存数据到disk的策略
#当有一条Keys数据被改变是，900秒刷新到disk一次
**save 900 1**
#当有10条Keys数据被改变时，300秒刷新到disk一次
**save 300 10**
#当有1w条keys数据被改变时，60秒刷新到disk一次
**save 60 10000**
#当dump.rdb数据库的时候是否压缩数据对象
**rdbcompression 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 /usr/local/redis/var/**

###########  Replication #####################
#Redis的复制配置
# slaveof <masterip><masterport> 当本机为从服务时，设置主服务的IP及端口
# masterauth<master-password> 当本机为从服务时，设置主服务的连接密码

#连接密码
# requirepass foobared

#最大客户端连接数，默认不限制
# maxclients 128
#最大内存使用设置，达到最大内存设置后，Redis会先尝试清除已到期或即将到期的Key，当此方法处理后，任到达最大内存设置，将无法再进行写入操作。
# maxmemory <bytes>

#是否在每次更新操作后进行日志记录，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。默认值为no
**appendonly no**
#更新日志文件名，默认值为appendonly.aof
#appendfilename
#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，always表示每次更新操作后手动调用fsync()将数据写到磁盘，everysec表示每秒同步一次（默认值）。
# appendfsync always
**appendfsync everysec**
# appendfsync no

################ VIRTUAL MEMORY ###########
#是否开启VM功能，默认值为no
**vm-enabled 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 logs/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
vm-page-size 32
vm-pages 134217728
vm-max-threads 4**

############# ADVANCED CONFIG ###############
**glueoutputbuf yes
hash-max-zipmap-entries 64
hash-max-zipmap-value 512**
#是否重置Hash表
**activerehashing yes**

注意：Redis官方文档对VM的使用提出了一些建议:
\*\* 当你的key很小而value很大时,使用VM的效果会比较好.因为这样节约的内存比较大.
\*\* 当你的key不小时,可以考虑使用一些非常方法将很大的key变成很大的value,比如你可以考虑将key,value组合成一个新的value.
\*\* 最好使用linux ext3 等对稀疏文件支持比较好的文件系统保存你的swap文件.
\*\* vm-max-threads这个参数,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.