*摘要: 本系列redis博客纯属个人学习总结记录之用，初学者，如若见不当之处，望指教。本篇是redis的两种持久化方式：rdb和aof方式*

  redis支持两种持久化的方式，可以单独使用或者结合起来使用

     第一种：RDB方式（redis默认的持久化方式）

 第二种：AOF方式

**一、RDB**

     rdb方式的持久化是通过快照完成的，当符合一定条件时redis会自动将内存中的所有数据执行快照操作并存储到硬盘上。默认存储在redis根目录的dump.rdb文件中。(文件名在配置文件中dbfilename)

**redis进行快照的时机**（在配置文件redis.conf中）

save 900 1：表示900秒内至少一个键被更改则进行快照。

save 300 10

save 60 10000

**redis自动实现快照的过程**

    1：redis使用fork函数复制一份当前进程的副本(子进程)

 2：父进程继续接收并处理客户端发来的命令，而子进程开始将内存中的数据写入硬盘中的临时文件

 3：当子进程写入完所有数据后会用该临时文件替换旧的RDB文件，至此，一次快照操作完成。

 注意：redis在进行快照的过程中不会修改RDB文件，只有快照结束后才会将旧的文件替换成新的，也就是说任何时候RDB文件都是完整的。 这就使得我们可以通过定时备份RDB文件来实现redis数据库的备份， RDB文件是经过压缩的二进制文件，占用的空间会小于内存中的数据，更加利于传输。

**手动执行save或者bgsave命令让redis执行快照。**

两个命令的区别在于，save是由主进程进行快照操作，会阻塞其它请求。bgsave是由redis执行fork函数复制出一个子进程来进行快照操作。

**文件修复：redis-check-dump**

**rdb的优缺点**

优点：由于存储的有数据快照文件，恢复数据很方便。

缺点：会丢失最后一次快照以后更改的所有数据。

**二、AOF**

     aof方式的持久化是通过日志文件的方式。默认情况下redis没有开启aof，可以通过参数appendonly参数开启。

**appendonly yes**

 aof文件的保存位置和rdb文件的位置相同，都是dir参数设置的，默认的文件名是appendonly.aof，可以通过      appendfilename参数修改

   appendfilename appendonly.aof

 redis写命令同步的时机

 a ppendfsync always 每次都会执行

 appendfsync everysec 默认 每秒执行一次同步操作（推荐，默认）

 appendfsync no不主动进行同步，由操作系统来做，30秒一次

**aof日志文件重写**

 auto-aof-rewrite-percentage 100(当目前aof文件大小超过上一次重写时的aof文件大小的百分之多少时会再次进行重写，如果之前没有重写，则以启动时的aof文件大小为依据)

 auto-aof-rewrite-min-size 64mb

**手动执行bgrewriteaof进行重写**

 重写的过程只和内存中的数据有关，和之前的aof文件无关。 所谓的“重写”其实是一个有歧义的词语， 实际上， AOF 重写并不需要对原有的 AOF 文件进行任何写入和读取， 它针对的是数据库中键的当前值。

**文件修复：redis-check-aof**

 **动态切换redis持久方式，从 RDB 切换到 AOF（支持Redis 2.2及以上）**

**CONFIG SET appendonly yes  
    CONFIG SET save ""（可选）**   
     **注意：当redis启动时，如果rdb持久化和aof持久化都打开了，那么程序会优先使用aof方式来恢复数据集，因为aof方式所保存的数据通常是最完整的。如果aof文件丢失了，则启动之后数据库内容为空。**

**注意：如果想把正在运行的redis数据库，从RDB切换到AOF，建议先使用动态切换方式，再修改配置文件，重启数据库。(不能直接修改配置文件，重启数据库，否则数据库中数据就为空了。)**