

RDB恢复机制

RDB是Redis的持久性策略的一种，Redis支持持久性，当Redis崩溃时，会导致内存中的数据都丢失，如果没有持久性，内存中的数据在磁盘中没有备份，这些数据就都丢失了，可能会丢失很重要的数据。

RDB就是持久性的一种策略，叫做RDB快照，具体策略就是手动或者自动将内存中的数据刷新到磁盘，生成一个当前时间节点的“快照”

手动触发

RDB有两种触发方式，一种是手动触发，一种是自动触发。其中手动触发又分为阻塞和非阻塞。

阻塞

阻塞的指令叫做save, 此时Redis就陷入阻塞，类似于keys *, 这是一个很危险的操作，因为Redis阻塞后不工作了，那么所有的读写请求就都到了磁盘数据库了，磁盘数据库就容易被击穿了。

非阻塞

非阻塞的指令叫做bgsave，此时Redis会创建一个后台进程，后台进程去将内存中的数据拷贝到磁盘形成快照。具体的做法就是如下：

Redis先判断一下，后台是否有进程正在更新RDB文件，如果有，说明正在更新RDB快照，返回即可。如果没有，那么开始更新RDB快照。首先调用fork创建子进程，子进程创建PCB、地址空间和页表。此时子进程内的数据和父进程的数据是一样的。然后子进程创建一个临时的快照文件，将内存中的数据拷贝到这个临时的快照文件之中，拷贝好了之后，就把这个新的临时快照文件替换掉旧的RDB文件。完成操作。

这里RDB是保存在Redis的工作目录中的。这个工作目录默认是/var/lib/redis。可以通过redis的配置文件进行修改和查看，配置文件目录为:/etc/redis/redis.conf。并且，如果是非阻塞手动触发bgsave，是子进程先创建临时文件，然后再替换；如果是阻塞save命令，那么就是直接在原本的RDB文件上面修改。

自动触发

常见的形成RDB快照的操作是自动触发。自动触发的方式有三种：

- 通过设置自动触发的条件，配置文件规定多长时间内，什么时候形成RDB快照。需要注意保存RDB文件是一个比较高的成本，所以不能太频繁。

- 当Redis正常退出时，即使用shutdown命令退出，会形成一个RDB快照。
- Redis进行主从复制时，主节点会生成RDB快照，然后将RDB快照传给子节点。

主要注意如果是kill -9或者异常崩溃，那么Redis会来不及保存快照数据，不会保存RDB文件。当RDB文件坏了，那么只有当Redis异常崩溃时RDB保持坏的状态，因为正常退出Redis会重新保存RDB快照。如果当RDB文件损坏后，Redis利用RDB文件恢复时，可能会启动失败。如果损坏的部分不影响启动，那么Redis能启动，此时数据有可能正确，有可能不正确。

配置条件

配置自动触发的条件是在/etc/redis/redis.conf中：

- save 900 1 每900s执行了一次redis操作，即保存RDB
- save 300 10 每300s执行了10次redis操作，即保存RDB
- save 60 10000 每60秒执行了10000次redis操作，即保存RDB

优缺点

优点

- 适用于进行全量复制，比如主从复制创建自己点的时候，适用于Redis全量数据进行恢复时进行操作。
- 同时RDB由于是二进制文件，所以数据进行恢复时，是比较快的，二进制数据，只需要将对应的数据直接放到内存中即可。

缺点

- 第一个缺点是有些老版本的RDB文件，不兼容新版本。就是说当我们的Redis进行升级时，可能旧版本的RDB文件就不能用来恢复Redis了。这个的解决方法就是遍历旧版本的Redis中的每一个key，然后将这些key插入到新版本的Redis中。
- 第二个缺点是不能实时化保存数据，在生成RDB快照之后，如果Redis再崩溃，那么从生成RDB快照到Redis崩溃这段时间中的数据就都丢失了。这个就需要用到AOF预写机制了。