

一.概念介绍

redis提供了两种持久化的方式，分别是RDB（Redis DataBase）和AOF（Apend Only File）。

RDB方式

RDB方式是一种快照式的持久化方法，将某一时刻的数据持久化到磁盘中。

- redis在进行数据持久化的过程中，会先将数据写入到一个临时文件中，待持久化过程都结束了，才会用这个临时文件替换上次持久化好的文件。正是这种特性，让我们可以随时来进行备份，因为快照文件总是完整可用的。
- 对于RDB方式，redis会单独创建（fork）一个子进程来进行持久化，而主进程是不会进行任何IO操作的，这样就确保了redis极高的性能。
- 如果需要进行大规模数据的恢复，且对于数据恢复的完整性不是非常敏感，那RDB方式要比AOF方式更加的高效。

AOF方式

AOF方式是将执行过的写指令记录下来，在数据恢复时按照从前到后的顺序再将指令执行一遍。

- AOF命令以redis协议追加保存每次写的操作到文件末尾.Redis还能对AOF文件进行后台重写,使得AOF文件的体积不至于过大.默认的AOF持久化策略是每秒钟fsync一次（fsync是指把缓存中的写指令记录到磁盘中），因为在这种情况下，redis仍然可以保持很好的处理性能，即使redis故障，也只会丢失最近1秒钟的数据。
- 如果在追加日志时，恰好遇到磁盘空间满、inode满或断电等情况导致日志写入不完整，也没有关系，redis提供了redis-check-aof工具，可以用来进行日志修复。
- 因为采用了追加方式，如果不做任何处理的话，AOF文件会越来越大，为此，redis提供了AOF文件重写（rewrite）机制，即当AOF文件的大小超过所设定的阈值时，redis就会启动AOF文件的内容压缩，只保留可以恢复数据的最小指令集。举个例子或许更形象，假如我们调用了100次INCR指令，在AOF文件中就要存储100条指令，但这明显是很低效的，完全可以把这100条指令合并成一条SET指令，这就是重写机制的原理。
- 在进行AOF重写时，仍然是采用先写临时文件，全部完成后再替换的流程，所以断电、磁盘满等问题都不会影响AOF文件的可用性。

二. 两种方式优缺点

1. RDB方式

•优点:

- 1.RDB是一个单一的紧凑文件,它保存了某个时间点得数据集,非常适用于数据集的备份,比如你可以在每个小时报保存一下过去24小时内的数据,同时每天保存过去30天的数据,这样即使出了问题你也可以根据需求恢复到不同版本的数据集.
- 2.RDB是一个紧凑的单一文件,方便传送, 适用于灾难恢复.
- 3.RDB在保存RDB文件时父进程唯一需要做的就是fork出一个子进程,接下来的工作全部由子进程来做, 父进程不需要再做其他IO操作, 所以RDB持久化方式可以最大化redis的性能.
- 4.与AOF相比,在恢复大的数据集的时候, RDB方式会更快一些.

•缺点:

- 1.Redis意外宕机,可能会丢失几分钟的数据 (取决于配置的save时间点)。RDB方式需要保存整个数据集, 是一个比较繁重的工作, 通常需要设置5分钟或者更久做一次完整的保存。
- 2.RDB 需要经常fork子进程来保存数据集到硬盘上,当数据集比较大的时候,fork的过程是非常耗时的,可能会导致Redis在一些毫秒级内不能响应客户端的请求.如果数据集巨大并且CPU性能不是很好的情况下,这种情况会持续更久。

2. AOF方式

•优点

- 1.使用AOF 会让Redis数据更加耐久: 你可以使用不同的fsync策略: 无fsync,每秒fsync,每次写的时候fsync.使用默认的每秒fsync策略,Redis的性能依然很好(fsync是由后台线程进行处理的,主线程会尽力处理客户端请求),一旦出现故障, 你最多丢失1秒的数据.
- 2.AOF文件是一个只进行追加的日志文件,所以不需要写入seek,即使由于某些原因(磁盘空间已满, 写的过程中宕机等等)未执行完整的写入命令,你也也可使用redis-check-aof工具修复这些问题.
- 3.Redis 可以在 AOF 文件体积变得过大时, 自动地在后台对 AOF 进行重写: 重写后的新 AOF 文件包含了恢复当前数据集所需的最小命令集合。 整个重写操作是绝对安全的, 因为 Redis 在创建新 AOF 文件的过程中, 会继续将命令追加到现有的 AOF 文件里面, 即使重写

过程中发生停机，现有的 AOF 文件也不会丢失。而一旦新 AOF 文件创建完毕，Redis 就会从旧 AOF 文件切换到新 AOF 文件，并开始对新 AOF 文件进行追加操作。

4.AOF 文件有序地保存了对数据库执行的所有写入操作，这些写入操作以 Redis 协议的格式保存，因此 AOF 文件的内容非常容易被别人读懂，对文件进行分析也很轻松。导出 AOF 文件也非常简单：举个例子，如果你不小心执行了 FLUSHALL 命令，但只要 AOF 文件未被重写，那么只要停止服务器，移除 AOF 文件末尾的 FLUSHALL 命令，并重启 Redis，就可以将数据集恢复到 FLUSHALL 执行之前的状态。

●缺点

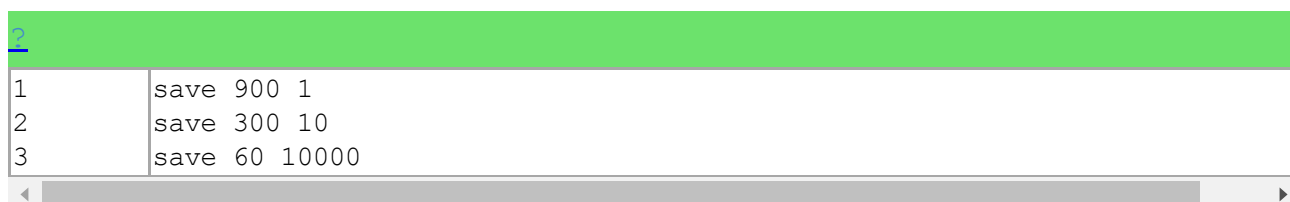
- 1.对于相同的数据集来说，AOF 文件的体积通常要大于 RDB 文件的体积。
- 2.根据所使用的 fsync 策略，AOF 的速度可能会慢于 RDB。在一般情况下，每秒 fsync 的性能依然非常高，而关闭 fsync 可以让 AOF 的速度和 RDB 一样快，即使在高负荷之下也是如此。不过在处理巨大的写入载入时，RDB 可以提供更有保证的最大延迟时间。

三. 配置方式

1. RDB配置方式

默认情况下，是快照rdb的持久化方式，将内存中的数据以快照的方式写入二进制文件中，默认的文件名是dump.rdb

redis.conf配置：



1	save 900 1
2	save 300 10
3	save 60 10000

以上是默认配置：900秒之内，如果超过1个key被修改，则发起快照保存；

300秒内，如果超过10个key被修改，则发起快照保存；

1分钟之内，如果1万个key被修改，则发起快照保存；

这种方式不能完全保证数据持久化，因为是定时保存，所以当redis服务down掉，就会丢失一部分数据，而且数据量大，写操作多的情况下，会引起大量的磁盘IO操作，会影响性能。

所以，如果这两种方式同时开启，如果对数据进行恢复，不应该用rdb持久化方式对数据库进行恢复。

2. AOF 配置方式

使用aof做持久化，每一个写命令都通过write函数追加到appendonly.aof中。

配置方式：启动aof持久化的方式

```
appendonly yes
```

参考资料：<http://redis.io/topics/persistence>

您可能感兴趣的文章:

- [Redis教程\(十\): 持久化详解](#)
- [内存型数据库Redis持久化小结](#)

原文链接：<http://www.jianshu.com/p/4a905a9f59a5>