预热篇

[**Redis学习手册(持久化)**](http://www.cnblogs.com/stephen-liu74/archive/2012/04/02/2365017.html)

Posted on 2012-04-02 07:44 [Stephen\_Liu](http://www.cnblogs.com/stephen-liu74/) 阅读(11188) 评论(4) [编辑](http://i.cnblogs.com/EditPosts.aspx?postid=2365017) [收藏](http://www.cnblogs.com/stephen-liu74/archive/2012/04/02/2365017.html)

**一、Redis提供了哪些持久化机制：**  
    1). RDB持久化：  
    该机制是指在指定的时间间隔内将内存中的数据集快照写入磁盘。      
    2). AOF持久化:  
    该机制将以日志的形式记录服务器所处理的每一个写操作，在Redis服务器启动之初会读取该文件来重新构建数据库，以保证启动后数据库中的数据是完整的。  
    3). 同时应用AOF和RDB。  
      
**二、RDB机制的优势和劣势：**  
    1). 性能最大化。对于Redis的服务进程而言，在开始持久化时，它唯一需要做的只是fork出子进程，子进程将快照数据写入到临时RDB文件中，当子进程完成数据写入操作后，再用临时文件替换老的文件。这样就可以极大的避免服务进程执行IO操作了。但是当数据集较大时，可能会导致整个服务器停止服务几百毫秒，甚至是1秒钟。  
    2). 相比于AOF机制，如果数据集很大，RDB的启动效率会更高。

3).系统一旦在定时持久化之前出现宕机现象，此前没有来得及写入磁盘的数据都将丢失。  
      
**三、AOF机制的优势和劣势：**  
    1). aof提供了3种同步策略，即每秒同步、每修改同步和不同步。Redis在每一次接收到数据修改的命令之后，都会将其追加到AOF文件中。在Redis下一次重新启动时，需要加载AOF文件中的信息来构建最新的数据到内存中。事实上，每秒同步也是异步完成的，其效率也是非常高的，所差的是一旦系统出现宕机现象，那么这一秒钟之内修改的数据将会丢失。而每修改同步，我们可以将其视为同步持久化，即每次发生的数据变化都会被立即记录到磁盘中。根据同步策略的不同，AOF在运行效率上往往会慢于RDB，每秒同步策略的效率是比较高的，每修改同步的方式在效率上是最低的。  
    2). 由于该机制对日志文件的写入操作采用的是append模式，因此在写入过程中即使出现宕机现象，也不会破坏日志文件中已经存在的内容。然而如果我们本次操作只是写入了一半数据就出现了系统崩溃问题，不用担心，在Redis下一次启动之前，我们可以通过redis-check-aof工具来帮助我们解决数据一致性的问题。  
    3). 对于相同数量的数据集而言，AOF文件通常要大于RDB文件。如果日志过大，Redis可以自动启用rewrite机制。即Redis以append模式不断的将修改数据写入到老的磁盘文件中，同时Redis还会创建一个新的文件用于记录此期间有哪些修改命令被执行。因此在进行rewrite切换时可以更好的保证数据安全性。  
      
**四、配置说明：**  
   1. Snapshotting:  
    缺省情况下，Redis会将数据集的快照dump到dump.rdb文件中  
    save 900 1              *#在900秒(15分钟)之后，如果至少有1个key发生变化，则dump内存快照。*  
    save 300 10            *#在300秒(5分钟)之后，如果至少有10个key发生变化，则dump内存快照。*  
    save 60 10000        *#在60秒(1分钟)之后，如果至少有10000个key发生变化，则dump内存快照。*  
  
      
   2. AOF配置：  
  Redis缺省的机制是RDB，如果需要使用AOF，则需要修改配置文件中的以下条目：  
    将*appendonly no*改为*appendonly yes*  
    appendfsync always     *#每次有数据修改发生时都会写入AOF文件。*  
    appendfsync everysec  *#每秒钟同步一次，该策略为AOF的缺省策略。*  
    appendfsync no          *#从不同步。高效但是数据不会被持久化。*  
      
   4. 如何修复坏损的AOF文件：  
    1). 将现有已经坏损的AOF文件额外拷贝出来一份。  
    2). 执行"redis-check-aof --fix <filename>"命令来修复坏损的AOF文件。  
    3). 用修复后的AOF文件重新启动Redis服务器。  
      
   6. Redis的数据备份：  
    在Redis中这是因为RDB文件一旦被生成之后就不会再被修改。Redis每次都是将最新的数据dump到一个临时文件中，之后在利用rename函数原子性的将临时文件改名为原有的数据文件名。因此我们可以通过copy的方式在线备份正在运行的Redis数据文件，在任意时刻copy数据文件都是安全的和一致的。创建cron job的方式定时备份Redis的数据文件，并将备份文件copy到安全的磁盘介质中。

加深篇

Redis数据备份与恢复

Redis提供了两种持久化选项，分别是RDB和AOF。

默认情况下60秒刷新到disk一次[save 60 10000 当有1w条keys数据被改变时]，Redis的数据集保存在叫dump.rdb一个二进制文件，这种策略被称为快照。

也可以手动调用Save或BGSAVE命令的：

/usr/local/bin/redis-cli -h 127.0.0.1 -p 6379 -a pwd bgsave

快照易恢复，文件也小，但是如果遇到宕机等情况的时候快照的数据可能会不完整。此时可能需要启用另一种持久化方式AOF，在配置文件中打开[appendonly yes]。

AOF刷新日志到disk的规则：

appendfsync always #always 表示每次有写操作都进行同步，非常慢，非常安全。

appendfsync everysec #everysec表示对写操作进行累积，每秒同步一次

官方的建议的everysec，安全，就是速度不够快，如果是机器出现问题可能会丢失1秒的数据。

也可以手动执行bgrewriteaof进行AOF备份：

/usr/local/bin/redis-cli -h 127.0.0.1 -p 6379 -a pwd bgrewriteaof

我们现在的做法是一主(Master)多从(Slave)只做了读写分离，Master负责写入和AOF持久化，Slave负责读取。每天通过 Slave做RDB备份[官方给的建议是每小时备份RDB文件，看你的策略和业务类型了]，并且会用脚本将相应的备份文件推送到备份服务器。

当redis服务器挂掉时，重启时将按照以下优先级恢复数据到内存：

•如果只配置AOF,重启时加载AOF文件恢复数据；

•如果同时 配置了RBD和AOF,启动是只加载AOF文件恢复数据;

•如果只配置RBD,启动是讲加载dump文件恢复数据。

恢复时需要注意，要是主库挂了不能直接重启主库，否则会直接覆盖掉从库的AOF文件，一定要确保要恢复的文件都正确才能启动，否则会冲掉原来的文件。