**【聊聊redis持久化 – RDB】**

**RDB方式，是将redis某一时刻的数据持久化到磁盘中，是一种快照式的持久化方法。**

**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。**

**虽然RDB有不少优点，但它的缺点也是不容忽视的。如果你对数据的完整性非常敏感，那么RDB方式就不太适合你，因为即使你每5分钟都持久化一次，当redis故障时，仍然会有近5分钟的数据丢失。所以，redis还提供了另一种持久化方式，那就是AOF。**

**【聊聊redis持久化 – AOF】**

**AOF，英文是Append Only File，即只允许追加不允许改写的文件。**

**如前面介绍的，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都执行一遍，就这么简单。**

**我们通过配置redis.conf中的appendonly yes就可以打开AOF功能。如果有写操作（如SET等），redis就会被追加到AOF文件的末尾。**

**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文件会变得越来越大，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，这点大家可以放心。**

**AOF方式的另一个好处，我们通过一个“场景再现”来说明。某同学在操作redis时，不小心执行了FLUSHALL，导致redis内存中的数据全部被清空了，这是很悲剧的事情。不过这也不是世界末日，只要redis配置了AOF持久化方式，且AOF文件还没有被重写（rewrite），我们就可以用最快的速度暂停redis并编辑AOF文件，将最后一行的FLUSHALL命令删除，然后重启redis，就可以恢复redis的所有数据到FLUSHALL之前的状态了。是不是很神奇，这就是AOF持久化方式的好处之一。但是如果AOF文件已经被重写了，那就无法通过这种方法来恢复数据了。**

**虽然优点多多，但AOF方式也同样存在缺陷，比如在同样数据规模的情况下，AOF文件要比RDB文件的体积大。而且，AOF方式的恢复速度也要慢于RDB方式。**

**如果你直接执行BGREWRITEAOF命令，那么redis会生成一个全新的AOF文件，其中便包括了可以恢复现有数据的最少的命令集。**

**如果运气比较差，AOF文件出现了被写坏的情况，也不必过分担忧，redis并不会贸然加载这个有问题的AOF文件，而是报错退出。这时可以通过以下步骤来修复出错的文件：**

**1.备份被写坏的AOF文件  
2.运行redis-check-aof –fix进行修复  
3.用diff -u来看下两个文件的差异，确认问题点  
4.重启redis，加载修复后的AOF文件**

**【聊聊redis持久化 – AOF重写】**

**AOF重写的内部运行原理，我们有必要了解一下。**

**在重写即将开始之际，redis会创建（fork）一个“重写子进程”，这个子进程会首先读取现有的AOF文件，并将其包含的指令进行分析压缩并写入到一个临时文件中。**

**与此同时，主工作进程会将新接收到的写指令一边累积到内存缓冲区中，一边继续写入到原有的AOF文件中，这样做是保证原有的AOF文件的可用性，避免在重写过程中出现意外。**

**当“重写子进程”完成重写工作后，它会给父进程发一个信号，父进程收到信号后就会将内存中缓存的写指令追加到新AOF文件中。**

**当追加结束后，redis就会用新AOF文件来代替旧AOF文件，之后再有新的写指令，就都会追加到新的AOF文件中了。**

**【聊聊redis持久化 – 如何选择RDB和AOF】**

**对于我们应该选择RDB还是AOF，官方的建议是两个同时使用。这样可以提供更可靠的持久化方案。**

**【聊聊主从 – 用法】**

**像MySQL一样，redis是支持主从同步的，而且也支持一主多从以及多级从结构。**

**主从结构，一是为了纯粹的冗余备份，二是为了提升读性能，比如很消耗性能的SORT就可以由从服务器来承担。**

**redis的主从同步是异步进行的，这意味着主从同步不会影响主逻辑，也不会降低redis的处理性能。**

**主从架构中，可以考虑关闭主服务器的数据持久化功能，只让从服务器进行持久化，这样可以提高主服务器的处理性能。**

**在主从架构中，从服务器通常被设置为只读模式，这样可以避免从服务器的数据被误修改。但是从服务器仍然可以接受CONFIG等指令，所以还是不应该将从服务器直接暴露到不安全的网络环境中。如果必须如此，那可以考虑给重要指令进行重命名，来避免命令被外人误执行。**

**【聊聊主从 – 同步原理】**

**从服务器会向主服务器发出SYNC指令，当主服务器接到此命令后，就会调用BGSAVE指令来创建一个子进程专门进行数据持久化工作，也就是将主服务器的数据写入RDB文件中。在数据持久化期间，主服务器将执行的写指令都缓存在内存中。**

**在BGSAVE指令执行完成后，主服务器会将持久化好的RDB文件发送给从服务器，从服务器接到此文件后会将其存储到磁盘上，然后再将其读取到内存中。这个动作完成后，主服务器会将这段时间缓存的写指令再以redis协议的格式发送给从服务器。**

**另外，要说的一点是，即使有多个从服务器同时发来SYNC指令，主服务器也只会执行一次BGSAVE，然后把持久化好的RDB文件发给多个下游。在redis2.8版本之前，如果从服务器与主服务器因某些原因断开连接的话，都会进行一次主从之间的全量的数据同步；而在2.8版本之后，redis支持了效率更高的增量同步策略，这大大降低了连接断开的恢复成本。**

**主服务器会在内存中维护一个缓冲区，缓冲区中存储着将要发给从服务器的内容。从服务器在与主服务器出现网络瞬断之后，从服务器会尝试再次与主服务器连接，一旦连接成功，从服务器就会把“希望同步的主服务器ID”和“希望请求的数据的偏移位置（replication offset）”发送出去。主服务器接收到这样的同步请求后，首先会验证主服务器ID是否和自己的ID匹配，其次会检查“请求的偏移位置”是否存在于自己的缓冲区中，如果两者都满足的话，主服务器就会向从服务器发送增量内容。**

**增量同步功能，需要服务器端支持全新的PSYNC指令。这个指令，只有在redis-2.8之后才具有。**

**【聊聊redis的事务处理】**

**众所周知，事务是指“一个完整的动作，要么全部执行，要么什么也没有做”。**

**在聊redis事务处理之前，要先和大家介绍四个redis指令，即MULTI、EXEC、DISCARD、WATCH。这四个指令构成了redis事务处理的基础。**

**1.MULTI用来组装一个事务；  
2.EXEC用来执行一个事务；  
3.DISCARD用来取消一个事务；  
4.WATCH用来监视一些key，一旦这些key在事务执行之前被改变，则取消事务的执行。**

常用命令

redis的常用命令主要分为两个方面、一个是键值相关命令、一个是服务器相关命令

1、键值相关命令

      keys \* 取出当前所有的key

      exists name 查看n是否有name这个key

      del name 删除key name

      expire confirm 100 设置confirm这个key100秒过期

      ttl confirm 获取confirm 这个key的有效时长

      select 0 选择到0数据库 redis默认的数据库是0~15一共16个数据库

      move confirm 1 将当前数据库中的key移动到其他的数据库中，这里就是把confire这个key从当前数据库中移动到1中

      persist confirm 移除confirm这个key的过期时间

      randomkey 随机返回数据库里面的一个key

      rename key2 key3 重命名key2 为key3

      type key2 返回key的数据类型

2、服务器相关命令

      ping PONG返回响应是否连接成功

      echo 在命令行打印一些内容

      select 0~15 编号的数据库

      quit  /exit 退出客户端

      dbsize 返回当前数据库中所有key的数量

      info 返回redis的相关信息

      config get dir/\* 实时传储收到的请求

      flushdb 删除当前选择数据库中的所有key

      flushall 删除所有数据库中的数据库